A Framework of Vision-Based Detection-Tracking Surveillance Systems for Counting Vehicles

A Thesis
Presented to
The Academic Faculty

By

Keitaro Kamiya

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Civil and Environmental Engineering

Georgia Institute of Technology

December, 2012

A Framework of Vision-Based Detection-Tracking Surveillance Systems for Counting Vehicles

Dr. Randall Guensler
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. Jorge Laval
School of Civil and Environmental Engineering
Georgia Institute of Technology

Dr. Michael Hunter
School of Civil and Environmental Engineering
Georgia Institute of Technology

Date Approved: November 9, 2012

ACKNOWLEDGEMENTS

I would like to thank my family, parents and colleagues in transportation laboratory, notably Mr. Chris Toth, Mr. Felipe Castrillon, and Mr. Evangelos Palinginis for their continuous support. I would also like to give special acknowledgement to Dr. Man-Woo Park and Dr. Ioannis Brilakis for their valuable advice and support throughout the conducted experiments. Lastly, I would like to thank the Georgia Department of Transportation (GDOT) for providing the video from Georgia NaviGAtor system.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLS OR ABBREVIATIONS

ITS	-	Intelligent Transportation Systems
GDOT	-	Georgia Department of Transportation
CCTV	-	Closed-Circuit Television Cameras
VDS	-	Vehicle Detection System
PDS	-	Presence Detection System
CMS	-	Changeable Message Signs
HoG	-	Histogram of Oriented Gradient
MoG	-	Mixture of Gaussian
ROI	-	Region of Interests
FoV	-	Field of View
F	=	Frame
t	=	Timestamp
l	=	Lane number
$v_{inst,F}$	=	Instantaneous velocity of a vehicle at timestamp *t*
$v_{avg,F}$	=	Averaged speed of a vehicle at timestamp *t*
$a_{avg,F}$	=	Averaged acceleration of a vehicle at timestamp *t*
$vr_{l,F}$	=	The estimated reference velocity at lane *l* at timestamp *t*
$D_{x,y}$	=	Real world coordinates x, and y of detected object's centroid
R_D	=	Quadrilateral region defined by the new detection
$T_{x,y}$	=	Real world coordinates x, and y of existing tracked object's centroid
R_T	=	Quadrilateral region defined by the existing tracked object
S_F	=	Scale factor for tracking for a vehicle at timestamp t
S_{ave}	=	Average scale factor of all present tracking objects at timestamp t
a_{th}	=	Threshold acceleration used for occlusion handling = $106 * 10^3 \, {}^{\text{miles}}/_{\text{h}^2}$
S_{th}	=	Threshold scale factor used for occlusion handling = 0.875
v_{th}	=	Threshold velocity value for high-pass filter = 20 mph

SUMMARY

This thesis presents a framework for motor vehicle detection-tracking surveillance systems. Given an optimized object detection template, the feasibility and effectiveness of the methodology is considered for vehicle counting applications, implementing both a filtering operation of false detection, based on the speed variability in each segment of traffic state, and an occlusion handling technique which considers the unusual affine transformation of tracking subspace, as well as its highly fluctuating averaged acceleration data. The result presents the overall performance considering the trade-off relationship between true detection rate and false detection rate. The filtering operation achieved significant success in removing the majority of non-vehicle elements that *do not* move like a vehicle. The occlusion handling technique employed also improved the systems performance, contributing counts that would otherwise be lost. For all video samples tested, the proposed framework obtained high correct count (>93% correct counting rate) while simultaneously minimizing the false count rate. For future research, the author recommends the use of more sophisticated filters for specific sets of conditions as well as the implementation of discriminative classifier for detecting different occlusion cases.

CHAPTER 1

INTRODUCTION

As computing power exponentially increases every year, the implementation of computer vision slowly became viable solution to many of the current problems faced within the transportation industry. Intelligent Transportation System (ITS) brings advanced computing, sensing, and telecommunication technologies to the transportation related problems. Specifically in the area of traffic detection, the public agencies and consulting firms relied upon magnetic loop detector installed inside the road surface for decades (Roess et al. 2004). This system however can be expensive in terms of installation cost, has finite coverage, and can only accommodate specific experimental purposes (e.g. counting, speed, etc.). To overcome the limitations of this system, imaging technology employing video cameras has been established as one of the most dominant tools assisting in the decision making of public agencies and traffic engineers.

Figure 1 illustrates typical commercialized software, placing virtual detecting zone on top of each traffic lanes in the video frames. Considering the change in pixel intensities, the detection algorithms initiates every time an oncoming vehicle crosses pre-determined detection zone while traffic volume or vehicle speed are produced simultaneously (Roess et al. 2004). For instance, speeds of the passing vehicle can be calculated based on the detector length and the time spent inside the detector (Tian et al. 2004). In the area of traffic monitoring, real-time acquisition of important traffic parameters such as traffic counts and traffic speed by time of a day allows traffic

engineers to make necessary operational changes to reduce congestion and promote safe driving.

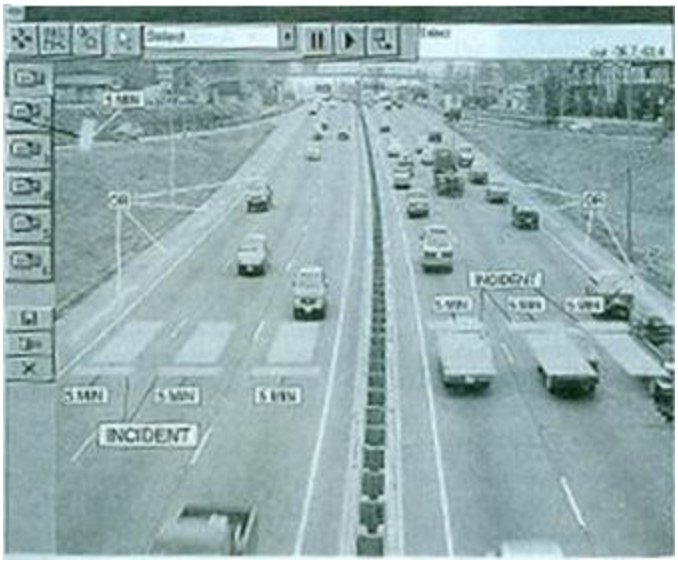


Figure 1: Typical Virtual Detectors, Source: (Roess et al. 2004)

The Georgia Department of Transportation (GDOT) is no different in relying on the new sensing technologies. GDOT took initiatives to establish a reliable ITS infrastructure for the purpose of traffic monitoring since 1996. As part of "Georgia NaviGAtor" systems, the current infrastructure now consist of 493 full-color Closed-Circuit Television cameras (CCTVs), 1,645 vehicle detection system (VDS) cameras, 39 presence detection system (PDS) cameras, and 116 changeable message signs (CMS) (David Ritchie et al. 2012). Typically, small monochrome or color electronic cameras are mounted on poles or bridges to record traffic conditions, and the output data are processed through various commercial imaging systems (Grant et al. 2000). While GDOT uses VDS data for measuring real-time traffic and road conditions such as average traffic speed and volume, there are always opportunities for improving the systems; following computational advances and technology.

Vision-based traffic surveillance has been one of the most promising fields for improvement and research. Still, many challenging problems remain unsolved, such as addressing vehicle occlusions and reducing false detection. Although the sensing technology provides overwhelming benefits, the stakeholders often forget that the detection of every vehicle in the video is extremely difficult considering changing environmental conditions such as illumination and occlusion. Many detection algorithms currently employed in the commercialized systems work well under ideal sets of conditions. However, many lack in adaptability to the dynamic nature of highway traffic. For example, the aforementioned VDS system reports as much as 74.2% error in counting vehicle volume at a site condition at night (Grant et al. 2000).

The majority of the existing or commercialized traffic counting systems (such as VDS) rely on computationally simple and fast detection algorithms applied on a localized region for a real-time feedback of counting data that can reflect the traffic condition. While this strategy is computationally effective and reasonably accurate, the system does not often address under-counting caused by occlusions, or the over-counting caused by false detection. Especially in the state of heavily congested traffic, incorrect detection can be severely exaggerated, because most of such systems rely on the motion-based detection which may count multiple overlapped vehicles as a single vehicle, or count the shadow casted by a vehicle as another vehicle. Conceptually, this type of error is caused by the lack of computational understanding of true motion of a vehicle (Figure 2). Unlike human beings, the computers have hard time segregating the sense of motion and identify the presence of occlusions by only using still images. Moreover, detection system alone

cannot differentiate detected objects in relation to those of past frames, thus assigning different vehicle IDs for all future detections (Figure 2).

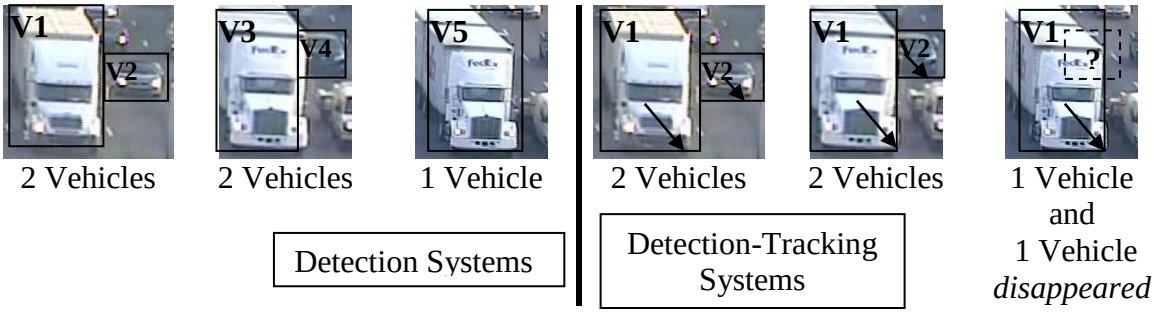


Figure 2: Comparison of Traditional Detection Systems with Detection-Tracking System

The use of detection-tracking system can offer distinctive advantages over aforementioned problems in traffic surveillance systems by providing a motion record of an object across subsequent video frames. First, given the positioning of a vehicle in 2-D road plane, frame-by-frame vehicle speed and acceleration computation is possible. These data can be useful for monitoring weaving and observing vehicle-to-vehicle relationship in term of speed differential and gap separation. Contrary to the traditional surveillance systems employing small limited detection zone, the proposed vehicle counting framework also allows a wider area of detection zones to be pre-set (as shown in Figure 3) to associate only reliable tracking data into the counting. The wider detection area allows more detection chance. This system is also capable of detecting a vehicle that would otherwise be missed in the traditional traffic counting methods, because even if the view of one object is hindered by another due to the occlusion at a particular video frame, a clear view of fully-occluded vehicle can typically be recovered before or after the full-body occlusion.

The system first applies background subtraction followed by Haar-like features detection trained with AdaBoost to detect the vehicle element as truthful as possible (Viola et al. 2001). Contrary to the motion-based detection, critical advantages of feature-based detection is its robustness against alignment error and such that more accurate counting per lane is allowed. The detected vehicle triggers the tracking algorithm which models vehicle with Eigen-images and estimate their location through particle filtering, and tracking data are converted into the real-world road coordinate enabled by a reliable calibration method.

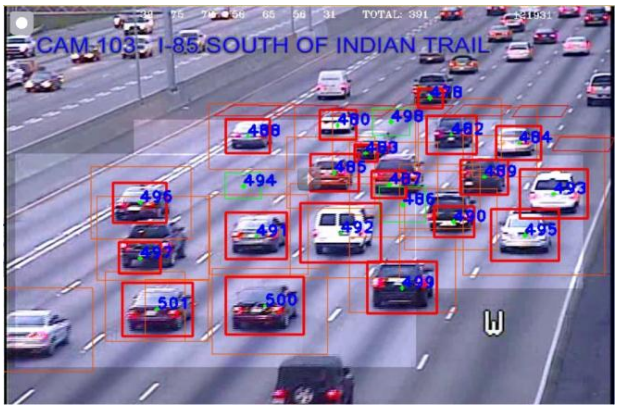


Figure 3: Wider Detection Zone as Indicated by White Transparent Area

As previously mentioned, the use of detection-tracking systems also allows iterative querying of each detected object's credibility by weighting the detected region not only based on its appearance, but also by its absolute and relative positions in the roadway in every video frame. The mathematical explanation of occlusion occurrence can be described as a function of a relative space difference between two (or more) motion vectors moving in the same dimensional space. For instance, typical occlusion cases result in relative displacement between two different detected regions approaching

toward each other where the movement of one vehicle based on the clear view may become uncertain during the occlusion. The inverse concept applies to the falsely detected non-vehicle, where the movement path of such object's trajectory has no clear relationship to the other stable motion vectors present.

Therefore in this thesis, the author aims to demonstrate that the typical problems affecting vision-based traffic surveillance systems, such as excessive false detections as well as the presence of occlusion can be recognized and mitigated by using the proposed detection-tracking framework. The presence of occlusions is detected by considering an abrupt change in speed of a vehicle and the shrinkage of tracking regions, taking advantage of incremental tracking algorithms with an ability to "remember" the past view of vehicle. Removal of non-vehicles is executed by considering the difference in the motion speed in the direction of traffic movement with respect to the rest of tracked vehicles present in each video frame. The video processing was conducted on the surveillance video data provided by Georgia NaviGAtor Systems in the state of Georgia, which covers most of Atlanta's highway corridors. The proposed method will be tested onto various 15-minutes long videos from different locations along the Georgia highway corridor under various viewpoint angles in respect to the road, illumination conditions, levels of congestion, and object size (pixels).

This thesis aims to demonstrate the effectiveness of the proposed detection-tracking framework in vehicle counting application by integrating vehicle motion. In Chapter 2, the specific research objective presented in this thesis is provided. In Chapter 3, background literature review is conducted and summarized introducing many of the existing methods concerning calibration, detection, and tracking. Chapter 4 provides

information regarding the overall system variables and methodology. Chapter 5 specifically focuses on the optimization of the detector performance relevant in proposed highway traffic setting. Chapter 6 presents the specific methodology and performance metrics that pertains to vehicle counting network. The results are presented in Chapter 7 as well as a detailed discussion of finding. The thesis conclusions and recommendations for future research are outlined in Chapter 8.

CHAPTER 2

RESEARCH GOALS AND OBJECTIVES

As mentioned in Chapter 1, stakeholders typically rely on VDS and other commercialized software for measuring traffic volume. While the data provided from this software are reasonably accurate in most cases, the system sometimes fails to cope under various conditions (e.g. extreme congestion). To overcome such difficulties, the detection-tracking framework is proposed in this thesis to demonstrate that the existing vehicle counting system can be improved by integrating vehicle motion into consideration.

The objective of this research is to construct a reliable framework for a detection-tracking system that can be useful to deal with wide varieties of current transportation related problems. Specifically in this research, a new vehicle counting framework is developed. The output results are compared against the manual count data for verification. The sample video streams are provided by GDOT surveillance cameras located throughout Interstate-85 highway in north-east Atlanta, Georgia. The research proposed in this thesis aims to demonstrate versatility and compatibility of a developed surveillance systems using the existing infrastructures and available technology.

In this detection and tracking framework, there are several unknown variables that could potentially affect the reliability of the proposed approach. First, the produced video sequences from the existing stationary cameras in Atlanta highway systems have different video qualities that make the detection phase difficult to succeed. Thus, the optimal setting of feature-based detection has to be determined empirically to understand

which set of pre-determined settings work best for detecting vehicles. The effectiveness of this template is carefully measured in the Chapter 5, considering different independent variables such as video quality, object size, feature direction, and amount of input data. By allowing wider detection zones compared to the traditional detection system, the optimally trained detector can accurately perform at its maximum potential.

Of those environmental variables, illumination conditions, object size, as well as the presence of shadow dynamically contribute to excessive over counting of traffic, because they often trigger false detection. The movement (i.e. change in position of an object in 2-D road plane) of these false detections provided by the detection-tracking system, however, often represent unusual travel path behavior in terms of its relationship to the instantaneous traffic state, making it unlikely to be a true vehicle. The proposed method aims to remove falsely detected non-vehicle objects by implementing high-pass filter in speed variance that is based on a simple traffic flow theory. This is done by keeping the average speed of last five tracked vehicles per lane as a reference speed, which is iteratively compared with all the potential vehicle candidate that were detected. If the speed of each candidate falls under the reference speed by same threshold, it is concluded that the detection is a false detection. The justification of chosen threshold value as well as relationship between the implemented filtering operations to the congested traffic (e.g. the speed of traffic is almost zero) are presented in chapter 7.

Lastly, because of the way the cameras are positioned alongside the highway, severe full-body occlusion has a significant impact on the tracking accuracy of the software. The proposed framework can detect and track each vehicle very well, even if the vehicle is partially occluded. However, full-body occlusion is sometimes imminent

when the clear view of vehicle is completely hindered by another larger vehicle in the adjacent lane. In such case, there is no way to capture the ground truth movement of a vehicle in a proposed monocular vision-based system unless some form of estimation algorithms are applied (e.g. trajectory estimation assuming that the vehicle retains the same speed prior to the occlusion). Conventionally, different occlusion handling has been demonstrated to be effective in vision-based frameworks in the past whose methodologies and implementation vary according to the used detection or tracking algorithms. Typically, partial occlusion handling is dealt with considering the visible features in the un-occluded part of vehicle (Fang et al. 2011; Kanhere et al. 2006). Zhang et al. (2008) considered the concavity of binary mask of detected region to identify the partial occlusion. However, such an approach can only be effective assuming that the detected region covers the whole view of vehicle, which may not be true for large vehicles. Moreover, typical occlusion handling is only designed to deal with partial occlusion and not with full-body occlusion, where clear view of an object is completely lost. In this particular framework, the occlusion handling can be implemented by taking advantage of employed tracking algorithm that can store the trajectory spatial data into temporary subspace, where the tracking failure can be detected in terms of vehicle's local view, such that if the scaling factor of each trackers diminish significantly per frame. By using this approach, the occlusion handlings can be triggered independent of information from the occluding vehicle. Additionally, averaged acceleration data is used on top of the scaling factor to distinguish potential false detection and true occlusion.

The overall effectiveness of the proposed method is weighted by the rate at which true vehicles are counted toward the counting zones (Correct Counting Rate: CCR), as

well as the rate at which non-vehicles are falsely counted (False Counting Rate: FCR) even after the implementation of proposed method. The removal rate (CRR) as well as the occlusion handling rate (OCR) is also shown to demonstrate the performance of both aforementioned processes.

2.1 Research Plan

By utilizing the proposed detection-tracking framework, the reliability and efficiency of the vision-based traffic surveillance systems can be further improved from the existing localized detection system. This research specifically focuses on improving the vehicle counting framework by incorporating the trajectory and speed of each detected vehicle, and allowing iterative querying of whether the detected object *moves* like a true vehicle. Some of the significant contributions presented in this thesis are as follows:

- The General detection-tracking framework is presented that can be used in wide variety of transportation related application.
- A tracking algorithm based on the 2-D road plane is presented such that any movements in the video frames are characterized by the direction of traffic movement, instead of simply using the 2-D image coordinate.
- The process of removing false detection excludes the counting of non-vehicle object if the speed of such object is too slow for the traffic state around the time of detection.
- Occlusion handling process is proposed combining tracking algorithm and traffic flow theory, considering vehicle movement that is unusual to typical monotonous vehicle movement in highway.

CHAPTER 3

LITERATURE REVIEW

3.1 Video Detection System

The critical advantage of computer vision technology is the ability to identify multiple objects simultaneously, which is possible by mean of conventional technologies. As a result, vision-based traffic surveillance systems have long been established as one of primary method to quantify traffic data, especially in this modern era where rapid growth of population has created an extremely dense and heavy traffic scene. Under an ideal field of view, many of the commercialized products (e.g. Autoscope, CCATS) being used today has known to provide the reliable data (Thi 2007).

3.2 Challenges of Existing Systems

In general, state-of-art commercialized products work by localizing the detection zone in a video to detect different types of vehicles (Mimbela et al. 2000). This system structure is computationally efficient and can run in a real-time basis while retaining very respectable detection rates. However, this kind of systems contain some inevitable problems caused by the object occlusion where larger vehicle with partially occluded smaller vehicle are typically considered as one object because foreground detection methods are not intrinsically designed to segregate multiple occluded vehicle (Malinovskiy et al. 2009). In another case, the appearance of larger vehicle or vehicle's shadow occluding the adjacent lanes also is known to trigger false detection (typically called "spillover" effect, Figure 4) in the commercial systems (Kanhere 2008).

Consequently, the merit of using computer vision as a surveillance tool has been limited by focusing strictly on building reliable systems that can perform in real-time.

Moreover, according to their descriptions or shown screenshot of an interface (Figure 4), most of the existing literature are performed at the ground-level observed from overpass bridge or high ground where detected vehicles size appear big with less noise relative to the video frame size . (Chaiyawatana et al. 2011; Kanhere et al. 2008; Negri et al. 2007; Hsieh et al. 2006). However, this is not the case for existing CCTV camera surveillance networks as most cameras are located very high up above the ground, and have an ability to pan, tilt, and zoom (PTZ camera). The practitioners presumably implemented this approach for two main reasons: 1) so that the view can cover the entire scheme of traffic condition with the least amount of stationed camera, and 2) so that the partial occlusions of multiple vehicles are less severe to deal with (Rodríguez et al. 2010). On the other hand, the most notable demerit of this approach is the size and qualities of detected object seem much less compared to if the camera was located at the ground level. This does not affect the detection capability of the system because the state-of-the-art background subtraction algorithms cope very well with motion variables, even if motion is minimal. However, degraded quality and smaller pixel-by-pixel objects may affect the tracking accuracy of vehicle objects because single pixel fluctuation in tracking quality in 2-D image coordinates results in greater difference in the estimated real-world vehicle trajectory. In previous literature, many empirically studied the feasibility of surveillance systems from the stationed roadway camera located high above the ground (Tsai et al. 2011; Bas et al. 2007; Grant et al. 2000) .

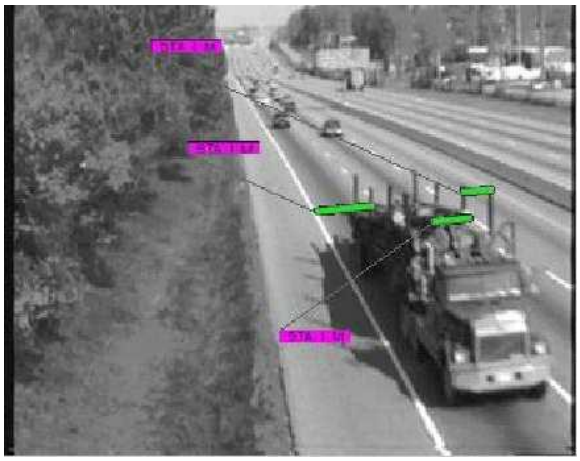


Figure 4: Example of Spillover Effect, Source: (Kanhere 2008)

3.3 Components of Detection-Tracking System

In general, individual objects are first recognized by the detection-and-tracking systems. (detection phase), specifying the pixel regions in an image where the objects will be present. For each subsequent video frame, the movement of initially detected regions are iteratively tracked and recorded through various tracking algorithms (tracking phase). Camera calibration is typically required to map the image coordinates onto real-world coordinate system so that any point in image can be converted to the real world dimension. In the following sections, general background information about detection-tracking framework commonly used in traffic surveillance systems is provided.

3.4 Background Subtraction

Motion-based detection considers a descriptive pattern of moving object separated from the background model to locate the interested objects. Detection is executed by comparing pixel-to-pixel information where the motion in an image is present. Many methods have been implemented in the application of vision-based traffic surveillance

tools: adaptive median filter (Sankari et al. 2011), color median (Zhang et al. 2007), frame differencing (Chaiyawatana et al. 2011), Mixture of Gaussian (MoG) (Bas et al. 2007; Tsai et al. 2011), wavelet differencing (Crnojevic et al. 2009), kernel-density estimation (Elgammal et al. 2002; Sheikh et al. 2005), and sigma-delta filter (Vargas et al. 2010). Chaiyawatana et al. (2011) employed frame differencing as well as thresholding comparing pixel intensities of vehicle objects to that of road. Tsai et al. (2011) used MoG models with shadow elimination based on color reflectance and gradient feature. Recently, a sigma-delta filter model was proposed by Vargas et al. (2010) to continuously recognize stopped vehicles as foreground objects while reducing computational cost. Traditionally, background subtraction model cannot cope with the intrinsic problems that are well-known in vision-based systems such as vehicle occlusion, shadow presence, illumination variance, camera shaking, and weather condition (Robert 2009). For example, several authors proposed a way to eliminate the shadow presence in the video by implementing additional processing on top of background subtraction (Huang 2010; Malinovskiy et al. 2009; Avery et al. 2007). However, the background subtraction model *alone* cannot provide reliable and practical surveillance systems because contemporary background subtraction by itself does not have the ability to precisely detect the vehicle objects at its center, which is crucially important for initiating accurate tracking record.

3.5 Feature-Based Detection

Many researchers have employed another distinctive approach, feature-based detection on top of computationally inexpensive background subtraction models to constrain the

regions in an image where the further processing will take place. This approach can mitigate the undercounting of vehicle because the detected window reflects the view of vehicles based on the trained template. As shown in Figure 5, the background subtraction (as indicated by large orange rectangle) may group multiple vehicles into one object, whereas feature detection can further separate the two vehicles (as shown by two red rectangles inside) based on their local appearance.

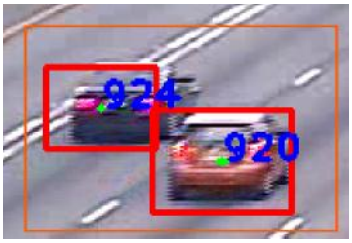


Figure 5: Background Subtracted Region and Detected Feature Region

In the past, various uniquely distinguishable features of vehicle objects have been targeted for this appearance-based approach, including, but not limited to: shape (Tsuchiya et al. 2006), edge (Malinovskiy et al. 2009), and symmetry (Liu et al. 2005). Some feature-based approaches also involve machine learning to learn the descriptive pattern (classifier) of an object class. During this process, the detection capability heavily relies on which visual features are selected for learning process. As such, simple raw data such as shape and edge are sometime inadequate to construct a robust detector. Further, point and edge information extraction in low-quality images containing heavy noise may be difficult because of lack of information, such as the correlation among neighborhood pixels. As shown in Figure 6, the correct edge boundary of a vehicle may be hard to visualize even to human eyes because of heavy noise in an image.

Figure 6: Enlarged View of Sample Vehicle from GDOT Camera

(Original Resolution: 46x33)

More descriptive pattern-based object tools such as Haar-like feature, Histogram of Oriented Gradient (HoG), and Local Binary Pattern (LBP) have proven to be effective for vehicle detection application. Haar–like features classify each object by comparing the intensity of every adjacent pixel values and differentiating them depending on the presence of intensity gradients (Cabrera et al. 2011; Feris et al. 2011). Haar-like features can be computed rapidly, and are known to succeed in processing low resolution images over other methodologies by considering a local intensities difference (Cabrera et al. 2011; Enzweiler et al. 2009). Recently, Haar-like feature without gradient information was trained with Support Vector Machine (SVM) to attain satisfactory result reaching 98% detection rate (Wen et al. 2007). Color-based Haar detection was also demonstrated to be effective contrary to traditional gray-scale Haar-like feature detection (Chang et al. 2007).

Another famous feature-based detection is the histogram of oriented gradient (HoG), widely used to consider gradient information and orientation around key point location of all pixels in vehicle image (Negri et al. 2007; Zhang et al. 2008). In Tamersoy et al. (2009), this methodology was further investigated by implementing unsupervised

learning across images collected by background subtraction. Benchmarking of Haar-like features and HoG features for vehicle detection was conducted in Negri et al. (2007), which were both demonstrated to be effective. In Zhang et al. (2007), point-based features called Multi-Block Local Binary Pattern (MB-LBP) are considered to detect flat areas, corner points and edges in the trained images using AdaBoost along with integral image for faster computation. In addition to the described benefit of appearance-based features above, the precise acquisition of centroid of vehicle object is now possible, because the detection window is appropriately fit to the vehicles' appearance size.

3.6 Tracking

Tracking of a vehicles trajectory can begin once an object is detected. Similar to detection methodologies, numerous tracking algorithms were implemented and improved in the traffic surveillance application, notably Kalman filter (Huang 2010; Bas et al. 2007; Hsieh et al. 2006), particle filter (Scharcanski et al. 2011), and mean-shift (Bouttefroy et al. 2008). Huang (2010) considered the corner points of detected vehicle when the eiganvalues of 2x2 gradient matrix exceeds a user-defined eiganvalue threshold (thus confirming that they are good feature points), and tracked using Kalman filter. There are some known hybrid models as seen in Negri et al. (2007) using projective Kalman filter combined with mean shift tracker. Others simply link independent detection in consecutive frames for tracking vehicles using a constant acceleration dynamic model (e.g. Tamersoy et al. 2009).

Many of the existing tracking strategies involve tracking of local feature points instead of capturing vehicle as a whole. This process is commonly known as feature

cluster tracking. Malinovskiy et al.(2009) considered Canny edge detection as well as Hough transformation was used to obtain the spatiotemporal information of the moving vehicle and those clustering Hough feature points are grouped together to represent the vehicle's travel path. However, assumptions are made that the vehicle always travel linearly (i.e. constant speed without movement perpendicular to the roadway), which is not always a real world phenomenon. Similar concepts are implemented by considering cluster of point-based feature that are tracked using Lucas-Kanade feature tracker provided in OpenCV (Kanhere et al. 2008). It is also important to recognize the importance of having an accurate travel path, as minor flickering movement of vehicle objects can cause great variability in instantaneous speed and acceleration. Although not transportation related, some researcher specializing in object tracking fields have successfully demonstrated the localized centroid update method can sufficiently alleviate small fluctuation in object movement (Mehmood et al. 2009). More descriptive information regarding some of the tracking algorithms presented in this chapter can be found in the document created by Yilmaz et al. (2006).

3.7 Camera Calibration

Even if the objects can be tracked successfully, it is impossible to construct reliable vehicle trajectory unless the width, length and height of the detected object have reference to the real-world dimensions. Thus, video calibration plays major role in providing meaningful data. Traditionally, this is done by manual calibration provided by the software (Mimbela et al. 2000). As shown in Figure 7, other researcher have approached camera calibration differently by considering spatiotemporal map using

perspective correction to map the camera view to 2-D top-down road view (Malinovskiy et al. 2009) In Hsieh et al. (Hsieh et al. 2006), the automated calibration was presented to estimate the width of traffic lanes based on the histogram of vehicle trajectory. The vehicles are first detected using difference image background subtraction and morphological operation, and tracked with Kalman filtering without any initial calibration. Then, the histogram of vehicle location can be constructed based on the recorded vehicle trajectory data, locating the driving lanes (Figure 8). However in this method, it was assumed that most vehicles move along the middle of individual lanes, which essentially hinders the quality of accurate tracking result.

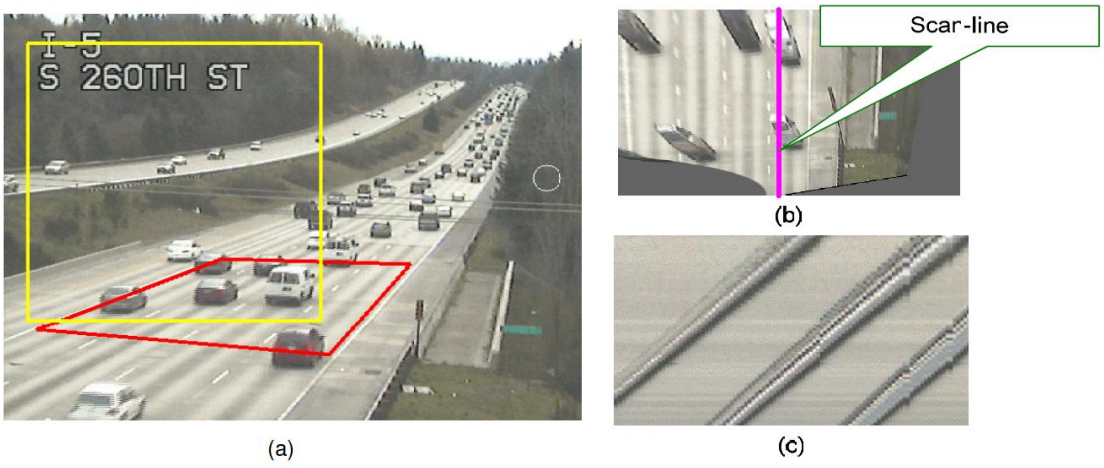


Figure 7: Perspective Correction

Source (Malinovskiy et al. 2009)

Figure 8: Lane Edge Extraction to Find the Potential Lane Width

Source: (Hsieh et al. 2006)

Many other implemented simplified world-to-image coordinate transformation assuming that the point to be tracked lie directly on top of roadway (i.e. distance in Z-direction = 0) (Kanhere et al. 2010; Rodríguez et al. 2010) This assumption is usually valid as the grade on highway does not abruptly change and level is typically flat. However, by using this approach, the individual vehicle height information is lost.

3.8 Summary

Now that background information pertaining to state-of-art detection systems, as well as the algorithms used for detection, tracking and camera calibration has been introduced, the framework of detection-tracking framework can be constructed, considering many of relevant algorithms provided in this chapter.

CHAPTER 4

GENERAL OVERVIEW OF A PROPOSED FRAMEWORK

The framework of detection-tracking is constructed considering many of the alternatives presented in the previous chapter and choosing the best options relevant to highway traffic surveillance systems. The basic methodology mainly consists of three stages; 1) calibration, 2) detection, and 3) tracking. First, the video samples are initiated in the software, and cameras' internal and external parameters are acquired through reliable calibration process proven to be effective in the highway surveillance application. Second, vehicles are recognized in the video stream automatically or can be manually initiated (detection phase) depending on the types of applications being used. Third, these detected vehicles are then handed over to the tracking (tracking phase). The vehicle counting framework provided in Chapter 6 employs the basic methodology provided here to improve the state-of-the-art counting systems. This chapter provides information about the general framework of the proposed systems that can be used in wide variety of transportation applications, such as trajectory analysis. On the other hand, a separate experimental study should follow up to this thesis to validate the accuracy of trajectory output for a feasibility study.

4.1 Camera (Coordinate) Calibration

The monocular CCTV cameras used in Atlanta highway provide a unique configuration where the view of scene is static and does not contain many miscellaneous objects other than vehicles. In this analysis, the roadway is assumed to be straight, with each lane

having a consistent width of 12 feet. By using monocular, fixed CCTV camera, a reliable calibration that relates 2-D image coordinate into 2-D road coordinate can be constructed that is known to be effective in the highway surveillance application (Kanhere et al. 2010).

To track the trajectory of vehicle movement in the later stage of this framework, each video sample must be calibrated to its unique environmental setting. For the purpose of this research, VLW (Vanishing point, Length, and Width) calibration strategy explained in Kanhere et al. (2010) has been implemented. First, the user manually draws two green lines along any traffic lanes on the video (Figure 10), which essentially define the vanishing point of the video by the intersection of these two lines. The yellow reference length is also manually drawn to indicate the converting dimensions (Figure 10) assuming that the distance between each lane being width of 12 feet and transverse length between each end of tick mark of dotted lane 40 feet (each tick mark is 30 feet apart, and line itself is 10 feet). The user provides the total width between two green lines and distance between each end of yellow reference length in a calibration window as shown in Figure 11. As a result, the transformation computes camera properties such as focal length (f), camera height (h), tilt angle (φ) (i.e. a vertical angle with respect to the ground plane, Figure 9) and pan angle (θ) (i.e. an angle formed with respect to road way lanes, Figure 9). Assuming that the camera has zero roll angles, square pixels, zero skew, and a principal points at the center of an image (Kanhere et al. 2010), only three parameters (e.g. f, h, and φ) are needed to construct transformation matrix. The remaining parameter θ maps the road coordinates with the direction of traffic flow (Kanhere et al. 2010).

Courtesy of Kanhere at el. (2010), the equations for finding f, h, φ, and θ are provided in Appendix A.

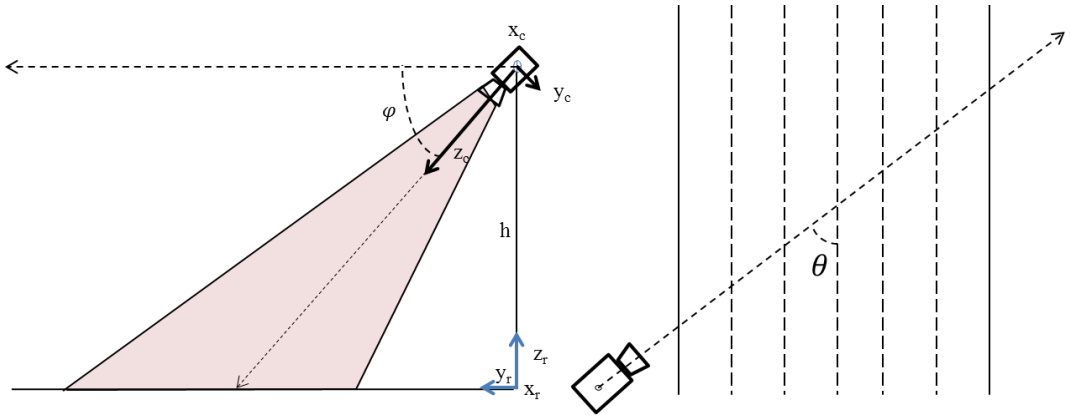


Figure 9: Side and Top View of a Camera Setup (Respectively)

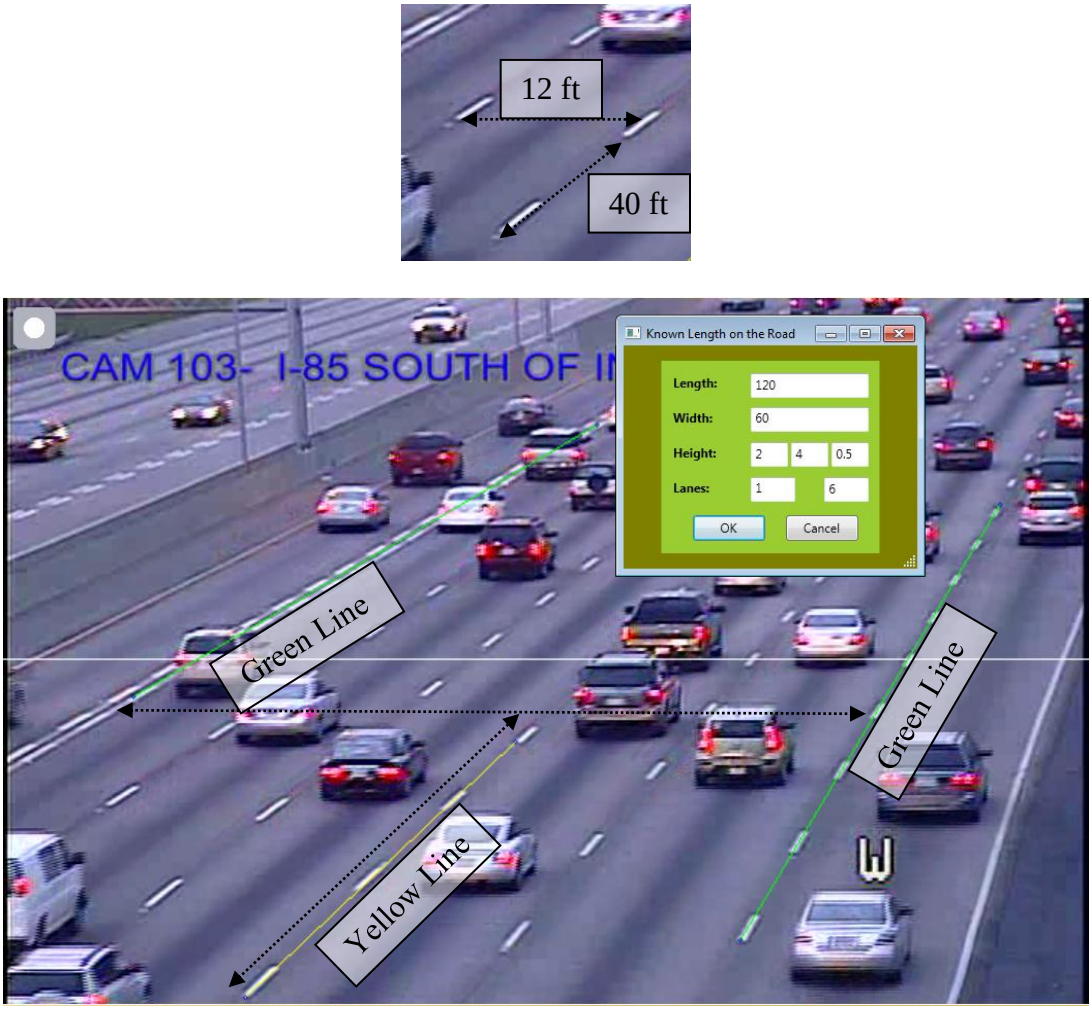


Figure 10: Calibration Process

The output data are used to construct a 3x4 homogeneous transformation matrix:

$$T_o = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin(\varphi) & -f\cos(\varphi) & fh\cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) & h\sin(\varphi) \end{bmatrix} \quad \text{(Eq. 2.1)}$$

which converts real-world three dimensional coordinate (x_r, y_r, z_r) into an image coordinate (μ_i, v_i) in following format.

$$\begin{bmatrix} \mu_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & -f\sin(\varphi) & -f\cos(\varphi) & fh\cos(\varphi) \\ 0 & \cos(\varphi) & -\sin(\varphi) & h\sin(\varphi) \end{bmatrix} * \begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} \quad \text{(Eq. 2.2)}$$

Assuming flat road, this equation further reduces to more simplified form when converting a point on the 2-D road plane to the 2-D image coordinate (i.e. substituting $z_r = 0$ in Equation 2.2 and simplifying).

$$\begin{bmatrix} \mu_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & -f\sin(\varphi) & fh\cos(\varphi) \\ 0 & \cos(\varphi) & h\sin(\varphi) \end{bmatrix} * \begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \quad \text{(Eq. 2.3)}$$

Notice that the size of transformation matrix reduces to 3x3. The above transformation, however, assumes that every point on the road is also a planer object, which is actually not true because vehicles have height in Z-direction. (Kanhere et al. 2010). To incorporate this change, the height of each individual vehicle further modifies transformation matrix T_o.

$$\begin{bmatrix} \mu_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & -f\sin(\varphi) & f(h-h_z)\cos(\varphi) \\ 0 & \cos(\varphi) & (h-h_z)\sin(\varphi) \end{bmatrix} * \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad \text{(Eq. 2.4)}$$

During the calibration process, the user specifies the range of height parameter for categorizing each vehicle height to the nearest values as shown Figure 11. This approximation is necessary to avoid redundant computation for transformation matrix for every single individual height with appropriate significant figures. The example shown in this figure, the height parameters are specified as initial height h_o, final height h_f, and increment value h_c. This information is used to create an array of 1xN size:

$$h_z \in [h_o, h_o+h_c, h_o+2*h_c, h_o+3*h_c, \ldots h_f]$$

The system approximates the height of individual vehicle to the nearest height value in an array. Subsequently, the elements in the array (h_z) are inserted into the transformation matrix T_o to create N number of different transformation matrixes each

corresponding to the different height parameters. The actual approximation of vehicle's height is described more in detail in section 4.3.2 of this chapter.

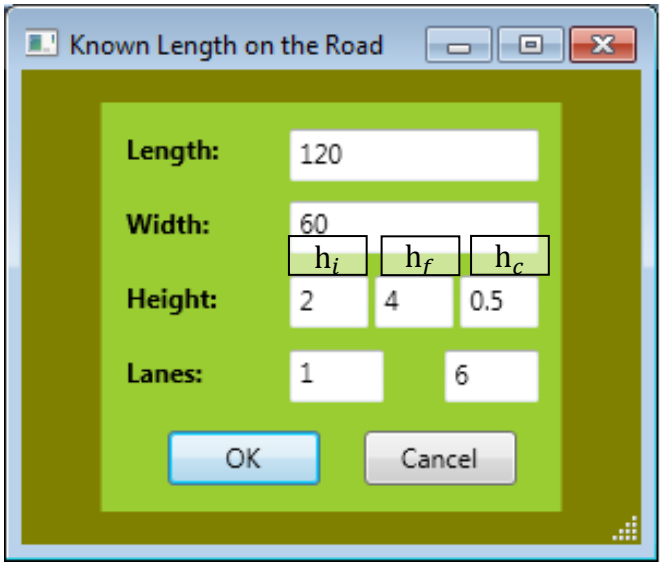


Figure 11: Calibration Window

The user also manually specifies the total numbers of lanes according to the video sample. In Figure 11, the number of the left specifies the number of lane to the left of green line, and the number on the right specifies the rest of lanes. Given the position of origin at the end edge of lane 0, the specified lane number identifies the location of tracking vehicle with respect to each lane based on the x-coordinate of the vehicle position and assumed 12 feet gap between each lane. For instance, the change in x-coordinate position of a vehicle from x = 11.90 feet to x = 12.10 feet could be used to identify the lane change from lane 0 to lane 1. The information provided in this step is crucial to the vehicle counting application described in Chapter 6.

Finally, the user interactions determine an area in the video called entry zone which defines a region in the video frame where the detection phase will take place

(Figure 12). The entry zone is typically drawn so that any stationary objects (such as the prescribed letters on the GDOT video) that can deter the tracking accuracy are not inside the area.

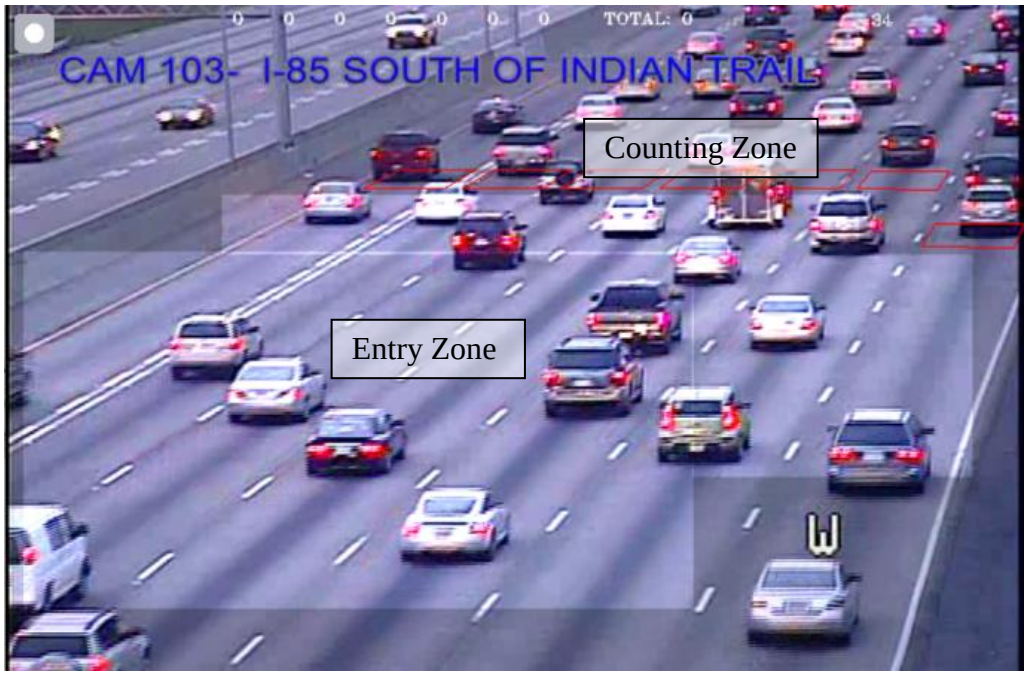


Figure 12: Entry Zone and Counting Zone

4.2 Detection Phase

The detection phase consists of two sequential algorithms, namely background subtraction and Haar-like feature detection (Haar-Cascade). The following two subsections describe these two more in detail.

4.2.1 Background Subtraction

Applying Haar-Cascade detection to every individual pixel element in the video frame is computationally intensive and should be avoided to reduce processing time. Applying Haar-Cascade to an entire image will also increase the likelihood of false detection, by

giving unnecessarily larger Region of Interest (ROI) where motion blurs may be present. For these two reasons, foreground feature (e.g. vehicle) is firstly recognized and separated from the background by the subtraction algorithm for easier processing and accurate result.

The OpenCV version of Mixture of Gaussian (MoG) background model is used throughout this thesis, which is thoughtfully described in notable papers (Cheung et al. 2004; Stauffer et al. 1999). Within the entry zone, the background subtraction results in several rectangle regions, each of which may contain one foreground feature. These small foreground blobs are accumulated to form one ROI where vehicles are more likely to be present. At the end, one enclosing rectangles are enlarged by some margin (e.g. 20 pixels in this thesis) for the proceeding Haar-Cascade method to extract a clear silhouette of a vehicle.

4.2.2 Haar-Feature Detection via Haar-Cascade

To detect each individual vehicle from the ROI, a method first proposed by Viola and Jones (Viola et al. 2001) is implemented; the open source code is available in a webpage OpenCV library. In this research, the code provided in EmguCV (a cross platform .Net wrapper to the OpenCV image processing library) has been implemented in the C#-based platform. The adaptive boosting classifier (AdaBoost) is used to train a Haar-Cascade template, which is computationally trained with both positive (vehicle) and negative (non-vehicle) images. The abundance of similar Haar-like features from the positive samples segregates the vehicle characteristics from that of negative images during

training. The detailed description and function used in the Adaboost-Haar algorithm are beyond the scope of this research but can be found in Viola et al. (2001).

4.2.2.1 Training Sample Collection and Training Process

A sufficient number of positive images and negative training images must be manually collected to generate a "detector" (otherwise known as a template). The discussions provided in Chapter 5 will investigate the training process in more detail.

4.2.2.2 Grouping of Multiple Detections

For the same vehicle within the Region of Interest (ROI) specified by the background subtraction, each Haar-Cascade is applied independent of each other. In other words, if a single vehicle is detected twice by two different windows of sub-region within ROI at one video frame, these two detections must be grouped as one vehicle. The default Haar-Cascade implementation from OpenCV takes in parameter called "minimum neighbor" whose value specifies the minimum number of overlapping detection for the object to be detected. Without grouping of neighboring detections, the counting will appear as shown in Figure 13, where multiple detections clearly represent single vehicles, but don't have a way to aggregate multiple detection regions based on the neighboring contents.

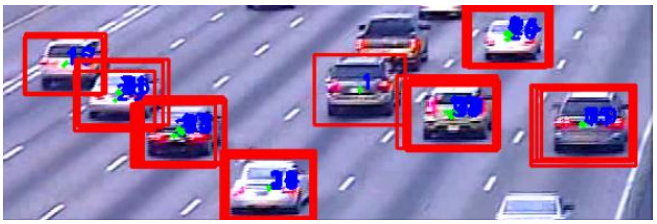


Figure 13: Detection without Grouping

In general, a detector recognizes more objects if the minimum neighbor value is lowered because of the reduced requirement on numbers of overlapping region to be detected. This does not mean, however, that the true detection rate increases. This is because reduction in minimum neighbor also increases the number of false detections (e.g. non-vehicle). For good practice, one may choose the middle-of-the-road minimum neighbor to maximize the true detection while keeping the false detection reasonably low. Figure 14 below describes the typical trade-off relationship between true detection rate false detection of any detector.

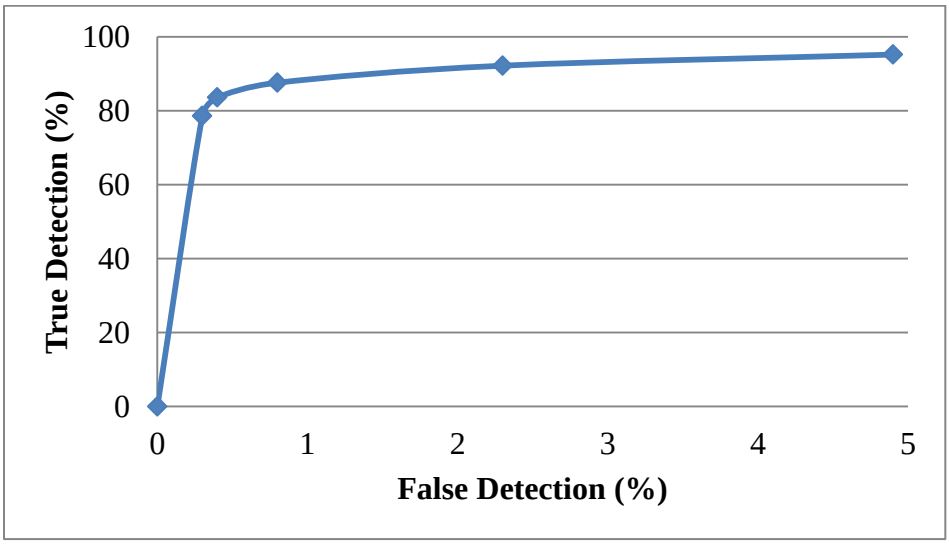


Figure 14: Typical Relationship between True Detection and False Detection

(Source: Table 4, Sample 1)

4.3 Tracking Phase

Each rectangular region resulting from the detection phase is initiated for independent tracking in the tracking phase. For a tracking algorithm, the method presented in Ross et al. (2008) is employed, in which the changes in the affine parameters of all future movement are determined by particle filtering. The centroid location (which is

determined by half of the width and length of each rectangle) in tracking region R_T, as well its initial affine parameters (i.e. rotation, scale, aspect ratio, and skew) are considered for tracking. The appearance of the vehicle is modeled with principal components of its eigen-images, which is stored in the temporal subspace that is repeatedly updated for each succession in tracking (Figure15). The implementation of this tracking algorithms works very smoothly with the Haar-Cascade detection used in the detection phase because the detection region establishes good starting point for initial eigan-image capturing without any further processing. As shown in Figure 15, this method is also fundamentally advantageous in the detection-tracking systems as each tracking elements have the sense of "things" being tracked rather than the cluster of identical characteristics often denoted by points or lines feature (Ross et al. 2008). The use of this tracking algorithm essentially allows a unique way to deal with problems that are specific to the traffic engineering application, such as detecting occluded vehicle by analyzing the abrupt change in the affine transformation of tracking regionR_T.

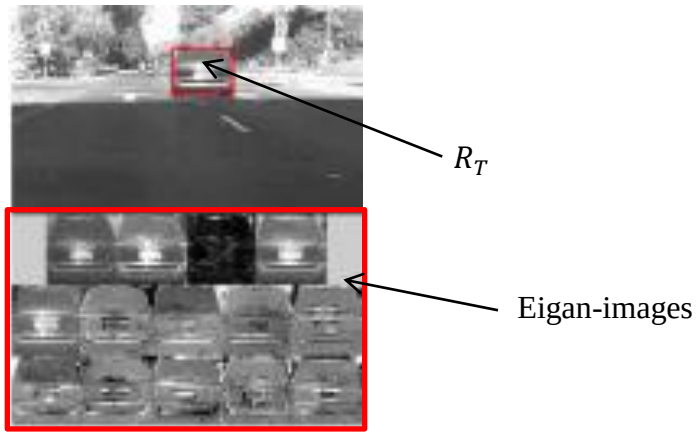


Figure 15: Tracking Region and Corresponding Eigan-images,

Source: (Ross et al. 2008)

4.3.1 Criteria for Keeping Newly Detected Vehicle

The iterative detection within the entry zones results in continuous detection of the vehicle tracked from the previous frames. The removal of these subsequent detections is thus necessary employing a similar concept used in Section 4.2.2.2 by considering the positional relationship among the local neighboring rectangles. If a detected region R_D results in a position that resembles the same tracked region R_T with high probability, the system discards the new detection. The first criterion compares the real-world positioning of two vehicles in the road coordinate assuming the horizontal safe clearance of 6 feet and transverse safe clearance of 18 feet, while the second criterion deals with relative Region of Interest (ROI) of two vehicles in the image coordinate considering the scale values. The function INFLATE(*region*, *factor*) simply inflates the input region by the input factor. If the newly detected object does not comply with either of the criteria, the new detection is categorized as a vehicle, and corresponding tracking is initiated. *Inf* factor of 0.3 is consistently used throughout the conducted experiments. The following is the criteria on this determination.

1) $(D_x\text{-}T_x)>6' \;\&\&\; (D_y\text{-}T_y)<18'$
2) $\text{INFLATE}(R_D, Inf) \in$ at least 2 out of 5 corner points of R_T (including its centroid point)

4.3.2 Vehicle Height Estimation

In this framework, the centroid of a detected region is chosen for representing the true movement of vehicles. As previously described in Section 4.1, estimating the virtual vehicle height is an essential part of successful coordinate conversion. Assuming that the

tracking region R_T always capture the whole view of a single vehicle and the bottom width of region R_T is on the road plane, the height estimation is executed considering the height of R_T and its aspect ratio. First, the real-world dimensions of vehicle width are estimated by taking the distance between bottom-right and bottom-left corner of R_T with height parameter equal to zero ($h_z=0$) in Equation 2.5. Considering an approximated proportional relationship between the distance in image coordinate and that of road coordinate, the height of a vehicle is estimated by the following equation.

$$\frac{\text{Width}_{\text{vehicle}}}{\text{Width}_{\text{vehicle}}} = \frac{\text{Height}_{\text{image}}}{\text{Height}_{\text{road}}} \quad \text{(Eq. 4.3)}$$

Here, the height of vehicle in image coordinate is given by half the height of R_T, or equivalently, the distance between the centroid and the bottom width of R_T (Figure 16). The converted vehicle height measured in feet falls in the range specified in Equation 2.5 and are rounded to the nearest incremental height h_z. Based on the value of h_z, the applicable transformation matrix T_o can be chosen for trajectory estimation of the centroid movement. This process is repeatedly applied to all tracking vehicles at all time frames.

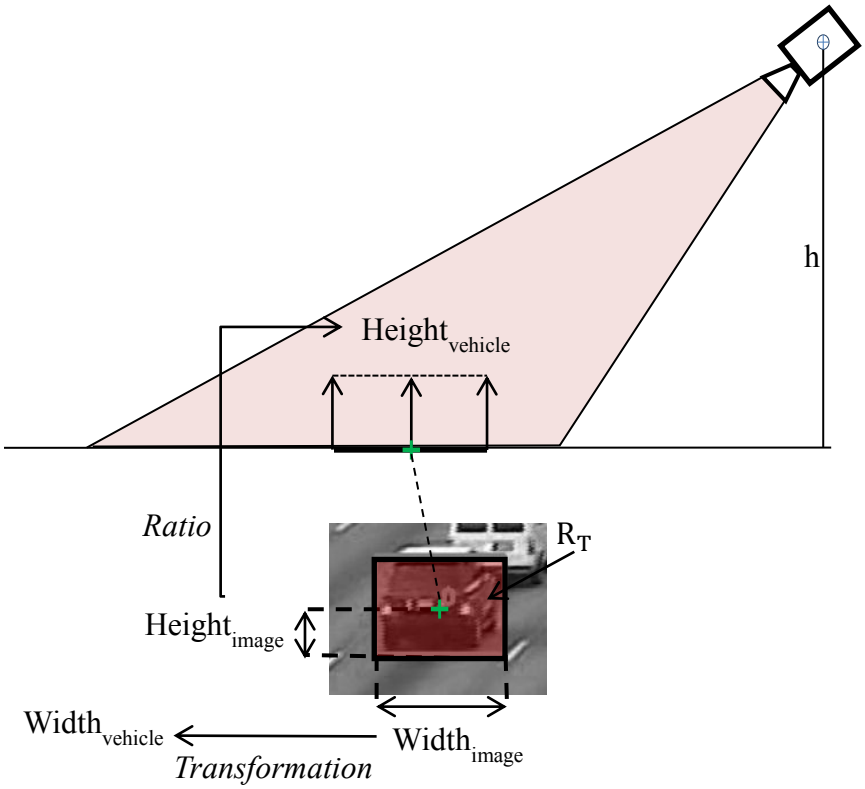


Figure 16: Vehicle Height Estimation

4.3.3 Velocity and Acceleration Calculation

Vehicle velocity and acceleration are estimated simply based on timestamps of input videos and transformed road coordinates of vehicles. Two types of speed are considered – instantaneous and average velocity, as well as an acceleration based on average speed.

$$v_{inst,F} = \frac{rw_F - rw_{F-1}}{t_F - t_{F-1}} \quad \text{(Eq. 4.4)}$$

$$v_{avg,F} = \frac{\sum\{v_{inst,F}, v_{inst,F-1}, v_{inst,F-2}, \ldots v_{inst,n}\}}{n} \quad \text{(Eq. 4.5)}$$

$$a_{avg,F} = \frac{v_{avg,F} - v_{avg,F-1}}{t_F - t_{F-1}} \quad \text{(Eq. 4.6)}$$

where: $v_{inst,F}$ = instantaneous velocity of a vehicle at a particular frame

rw_F = 2-D position in road plane

t_F = timestamp of a frame

$v_{avg,F}$ = averaged velocity of a vehicle at a particular frame

n = amount of frames approximately equivalent of one second

$a_{avg,F}$ = averaged acceleration of a vehicle at a particular frame

In the majority of applications considered in this framework, the use of average velocity is recommended rather than instantaneous velocity based upon frame-by-frame estimation because several video samples used in this research contain timestamp noise. In Figure 17, the sample data for one vehicle provides the figure illustrating disparities between two velocities. Hence, constant *n* in Equation 4.5 actually varies for all frames because of the inconsistency in the timestamp. Thus, *n* value is normalized based on local frame rate covering approximately one second rather than using constant number of frames for average velocity calculation. The acceleration data calculated from averaged speed can provide a valuable information, but its value fluctuates significantly because 1) the frame rate of each video varies, and 2) since the timestamp for each frame is significantly small (often ~0.05 second per frame or less), small difference in the average speed have a large impact in the corresponding acceleration data. In this experiment, the method provided in Chapter 6 only considers acceleration data as a metrics to observe and differentiate vehicle-like movement. On the other hand, the actual comparison between the output acceleration data with the ground-truth acceleration of a vehicle was

not explicitly considered in this thesis. In the future, adequate field studies will follow up to this research to understand the feasibility and reliability of the acceleration data.

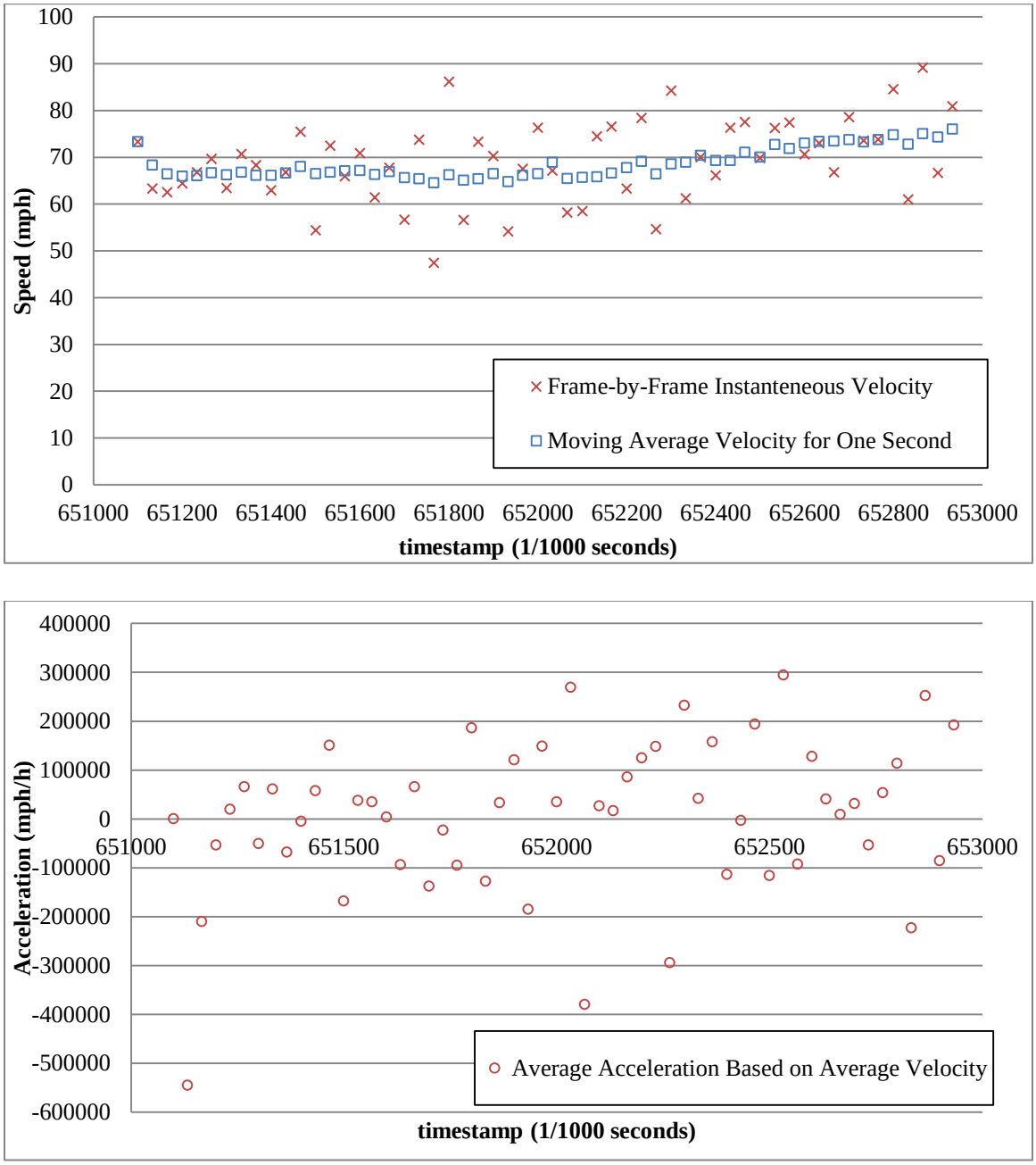


Figure 17: Example of Instantaneous Velocity vs. Average Velocity (Top), and Average Acceleration (Bottom)

CHAPTER 5

OPTIMIZATION OF HAAR-CASCADE

Ideally, the detection template should be created based on the positive images corresponding to the specific sets of environmental variables (e.g. illumination, camera angles, etc.) shown in each videos. However, this is not practical because the manual labor costs required to collect enough data to construct a reliable detector under all possible conditions are a significant burden to the practitioner (Last 2007). To demonstrate the versatility of utilizing feature-based templates and minimizing labor costs, only one template for each direction (e.g. toward the camera and away from the camera) is constructed and used throughout the experiments.

Haar-Cascade implementation used in this thesis research is typically capable of recognizing various object features with relative ease and simplicity. However, very little is known regarding the kind of intrinsic characteristics of Haar-like feature detection algorithms that can impact the detection result in a positive ways, especially in a real-world application where experimental procedures are less flexible and favorable experimental set-ups are impossible. The cameras and captured video streams are all relatively different among each other in terms of image quality, frame rate, ability to zoom-in close and out far, etc. Thus, in this chapter, various experiments are conducted to assess the optimized detection within controllable range. The tested parameters include image quality, object size, feature detection, and amount of input data.

5.1 Training Process

Detector templates are created by applying an adaptive boosting classifier, which compiles a sequence of weak detector to form one strong detector. The classifier requires positive samples (i.e. image of vehicles) that represent the targeted objects, and negative samples (i.e. any image that does not contain a vehicles) to differentiate the two. Both samples are reformatted to much smaller object size during the training process (24x18 pixels for rear view, and 22x16 pixels for front view) which retains an average aspect ratio across all samples.

5.1.1 Positive Samples

The positive samples represent the objects of interest; in this case a collection of vehicle images that are input to the systems. The positive samples are the most essential component of the learning process, as the fundamental algorithm used in Haar-Cascade relies on the pixel gradient (or intensity) in the vehicle images. Positive samples used in this thesis consist of small to medium size vehicles, such sedans, SUVs, pick-up trucks, vans, etc. Any other vehicle types such as motorcycle and large heavy-duty trucks may require more extensive descriptor and were beyond the scope of the research. Figure 18 shows the sample positive images for both front and rear view.

5.1.2 Negative Samples

Negative samples describe a collection of images that do not represent the objects of interest. Figure 18 shows the sample negative images.

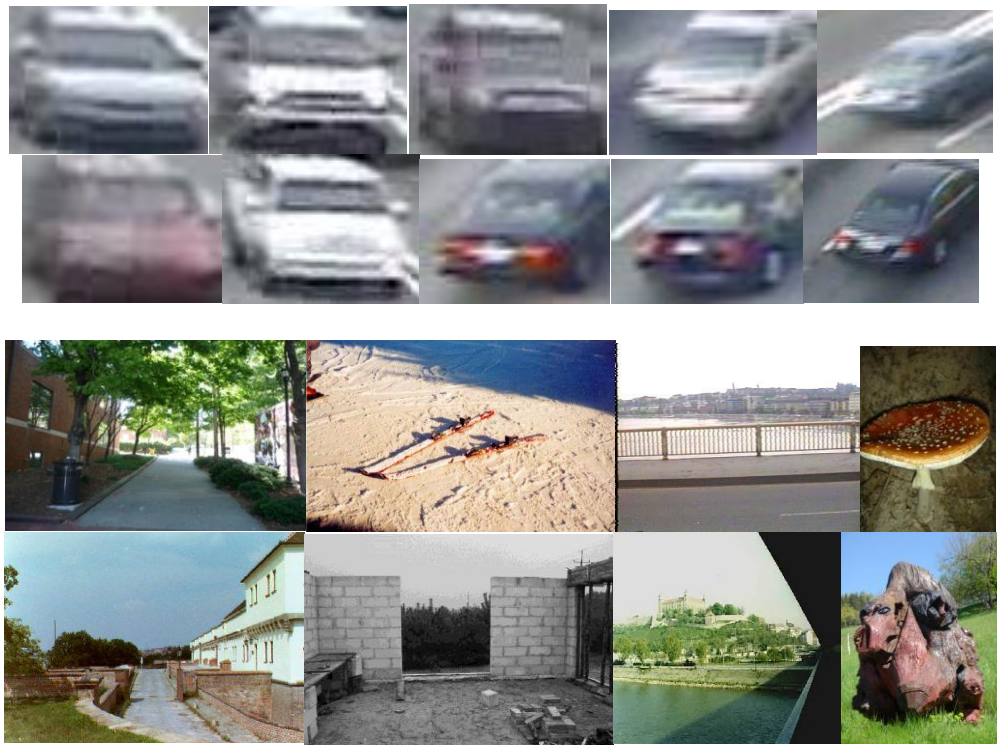


Figure 18: Positive (Top) and Negative Sample (Bottom) Images

5.2 Testing Parameters

The following subsections describe the testing parameters that can lead to optimization of Haar-Cascade.

5.2.1 Image Quality

The video samples obtained from GDOT CCTV cameras contain varying image quality that result from low resolution original images and quantization effects during the format conversion. The original videos must be converted to the allowable video format, typically .mp4 or .avi extension for the software to work. The better quality images are expected to result in a better detecting performance after the machine learning process

because of more apparent feature space that can be more easily detectable by Haar-Cascade without large amounts of noise.

5.2.2 Object Size

Even if the camera location is the same, the Field of View (FoV) for two recordings may be different, depending on the zooming and rotation setting of PTZ camera. This essentially causes the same vehicle object to appear with a larger or smaller resolution. This fact however should not affect the detection accuracy, as the initial templates are trained with much smaller object size (e.g. 24x18 pixels). For example, Enzweiler et al. (2009) reports that the Haar wavelet features performs well in low resolution pedestrian images (18x36 pixels). On the other hand, choosing higher resolution images for training may cause the algorithms to focus on too specific object properties, increasing the computational load rapidly (Haselhoff et al. 2008).

5.2.3 Feature Direction

The original Haar cascade detection proposed by Viola et al. (2001) dealt with recognizing only horizontal and vertical Haar-like features in the image by using the integral image algorithms. Later, Lienhart and Maydt (2002) proposed an algorithm that can consider 45 degree Haar-like feature recognition in addition to Viola's work. The EmguCV implementation of Haar-Cascade can accommodate both configurations specified by “ALL” (horizontal, vertical and diagonal features) and “BASIC” (only horizontal and vertical haar features) parameters during the training initialization. Figure 19 shows the example of Haar feature directions where the black and white rectangles each corresponding to the detected feature based on the difference in the local pixel

intensities. As displayed in the same figure, the diagonal 45 degree features may not correctly match up with the corresponding feature on the vehicle image because each vehicle may appear differently to the camera with respect to the perspective angles.

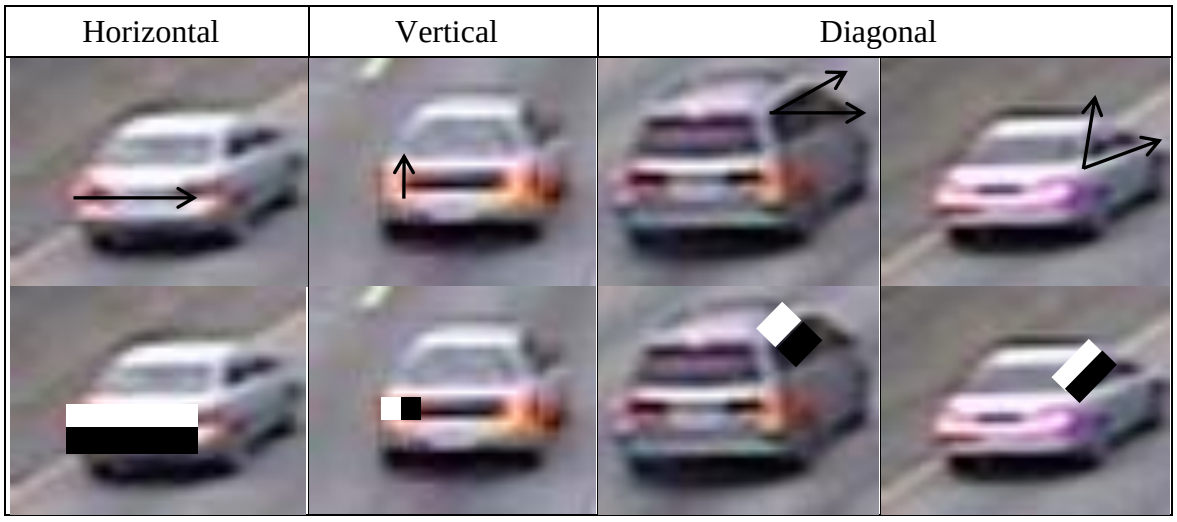


Figure 19: Example of Haar Feature Direction

5.2.4 Number of Positive Samples for Training

As the amount of data used for training increases, the detector typically becomes more robust in terms of providing accurate detection result. However, an article by Boonyanunta et al. (2004) reports that the robustness of detector reaches a definite limit where any further increase in input data does not necessarily improve the detector. In this research, experiments were conducted to observe this relationship and provide an estimate of an optimal number of samples that should be used from the GDOT camera. To control the experiments, the negative samples used in this experiment were kept constant and only positive samples were varied.

5.3 Experimental Results

Table 1 shows the comparative results for considering BASIC or ALL features that were tested in three different locations with one of the three tested with two different training sets. The result indicates that the BASIC features that consider only horizontal and vertical features detect true vehicles than that of ALL features that adds 45 degree diagonal features. This result is somewhat counter-intuitive as the detection rate worsened as more feature directions are considered during the training. However, the use of Haar-Cascade in the highway traffic observed from static monocular camera offers a unique view of environment where there is abundance of horizontal and vertical directional vectors along the face of vehicles that are almost identical for every vehicle, which facilitates greater confidence in the creation of robust detector. By incorporating additional 45 degree directional vectors, the classifier does its best to aggregate the data from all samples, which may in fact introduce additional random features into the systems, because even in the same videos, the angles of its diagonal features are not aligned for all vehicles (Figure 19). The pictures within Table 1 also confirm the hypothesis that the object sizes of vehicle in the video does not seemed affects the detectability, which was expected.

Table 1: Detection Rate for Varying Feature Directions

Cam 212 Roswell	Positives	Detection Rate
Using All feature	600	67%
Using Basic feature		86%

Cam 101	Positives	Detection Rate
Using All feature	632	69%
Using Basic feature		72%
Using All feature	628	17%
Using Basic feature		76%

Cam 42	Positives	Detection Rate
Using All feature	200	73%
Using Basic feature		90%

Diagonal feature computation also involves allocating the edge information to fit the given window size, which may bring an approximation round-off error. Messom et al. (2006) also reports that the use of rotated feature in general is impractical because of the rounding error to approximate the width and height of rotated rectangle region to fit the pixel boundaries. As shown in Figure 20, Haar-like feature classification may provide complete different rectangular region depending on the starting location of line information.

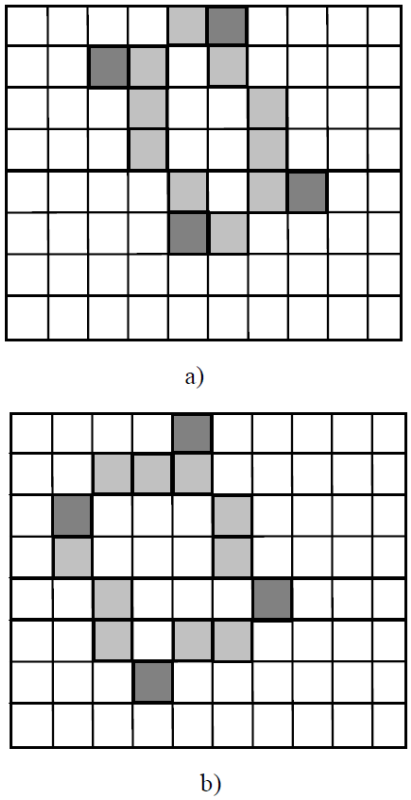


Figure 20: Round off Error during the Approximation of Feature Direction, Source: (Messom et al. 2006)

In Table 2, detection rate comparison between the template trained with original and bad quality images are presented. The result seems to insinuate that the detection performance is better if the template is based on the original quality image, which corresponds to the constructed hypothesis.

Table 2: Detection Rate for Bad and Original Quality Images

Location: Cam 101	Detected Count	Manual Count	Detection Rate
Original Quality	847	957	88.5 %
Bad (quantized) Quality	760	957	79.4 %

Ultimately, based on these preliminary results, more comprehensive testing was conducted, comparing the result of training based on (1) bad images and ALL features, and (2) original image and BASIC features. Figure 21 and 22 below shows the detection rate as a function of positive samples to be trained for both image sets. For both figures, positive samples are randomly extracted for each iteration from the same sample pool (2181 front view vehicles). For practical consideration, random pool of 200, 400, and 800 image samples are selected 10 times, while for the image set (2), Another five sets of templates were constructed at 1200, 1600, and 2000 images. As a result, the detector trained with image set (1) can only detect up to 62% detection rate on average. Also, the detection rate is widely destributed as shown in standard deviation bar. On the other hand, the image trained with image set (2) demonstrate that on average, the detection rate can almost reach 95% with just 400 positive samples.

It is also important to note that for both figures, the graphs seem to approach the maximum detection threshold that is explained earlier in this chapter, which corresponds to the conclusion the past researcher reached in the field of data (Boonyanunta et al. 2004). Especially for the image set data (2), the detection rate increase from 400 images to 2,000 images are merely 3% at a cost of accumulating 1,600 more image sets. This results insinuate that understanding how the training curve behave for each video sample sometimes can be crucial in practice because an additional labor cost required to gain small marginal benefit in the detection rate can be avoided.

Nevertheless, the result suggests that the relative detector performance is much better when using BASIC features and original images, rather than ALL features and bad images. Thus, for the template used in the following chapter, the template are created

based on the optimized configuration such that only horizontal and vertical features are considered while using the original quality images.

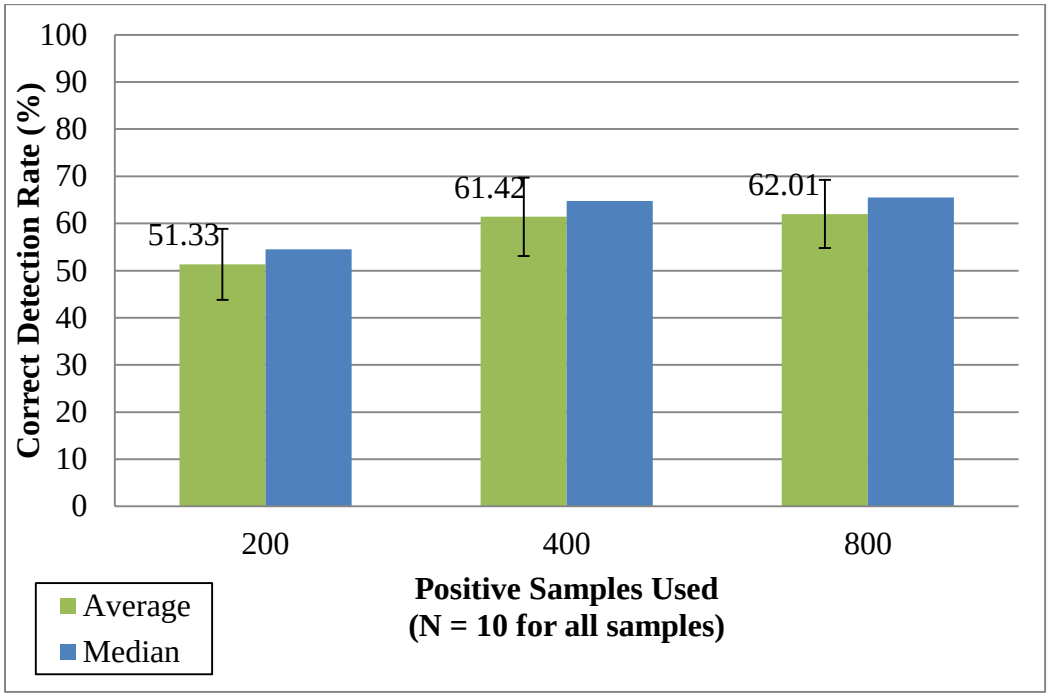


Figure 21: Detection Rate of ALL Features (Horizontal, Vertical, and Diagonal) and Bad Quality Image Sets

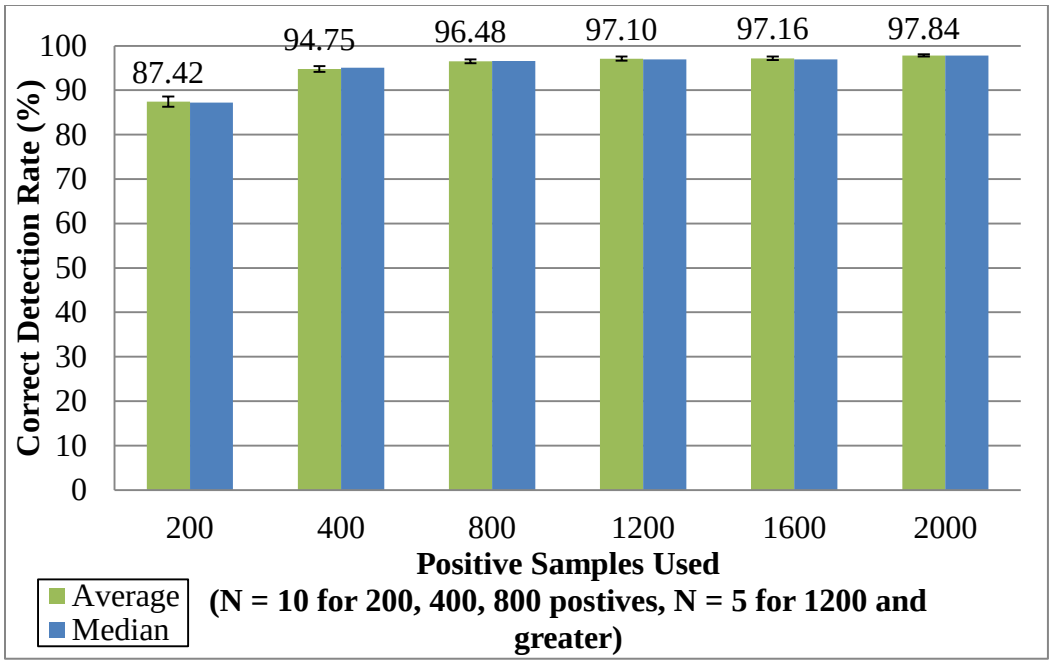


Figure 22: Detection Rate of BASIC Features (Only Horizontal and Vertical) and Original Quality Image Sets

5.4 Summary

This chapter provided basic Haar-Cascade implementation as well as the optimization of its detection parameters for the images collected from GDOT cameras. Findings reported in this chapter correspond to the past literature regarding the diagonal features, object size, and detector learning performance. Based on the result, use of BASIC features with original quality image for training is recommended. The detection-tracking framework can ultimately employ the determined optimal parameter for detecting vehicle.

CHAPTER 6

METHODOLOGY FOR COUNTING VEHICLES

6.1 Motivation

In this chapter, the vehicle counting application is presented using the proposed framework. As described in Chapter 3, localized detection in a limited region cannot effectively cope with the false detection of vehicles (i.e. detection of non-vehicle, such as road or shadow) or the occlusion of multiple vehicles. The methodology presented in this chapter is constructed to minimize problems common to vision-based surveillance systems by incorporating the tracking algorithm and general traffic theory.

Figure 23 shows the overall framework of this application. The sample videos are first introduced to the systems and calibrated by the method provided in Chapter 4. Then, the entry zone is specified by the user, allowing maximum regions where detection or tracking works the best such that the entry zone does not contain any stationary objects that would hinder the tracking capability. The video is segmented into its sequence of images with timestamp *t*, and applied under the background subtraction algorithms and Haar-Cascade feature detection to detect any new vehicle in the scene. For every vehicle detected, the tracking algorithm tries to capture its movement in the x-y plane (i.e. 2-D road plane) for all future subsequent frames as long as the vehicle remains in the entry zone. The tracking simultaneously updates the trajectory record of same vehicle, followed by height estimation, coordinate transformation (i.e. transformation from 2-D image plane to 2-D road plane), and velocity and acceleration. Additionally, movement

of each vehicle depicted by the tracking iteratively queries for identification of false detection and occlusion in all future frames to avoid undercounting or overcounting. Occlusion identification is prioritized before the false detection identification. As soon as the vehicle is recognized as occluded, the presence of that vehicle is carried over until the counting zone even if the view of a vehicle is completely lost. In a case of slow moving traffic, the occlusion handling simply removes the vehicle from tracking to rely on subsequent re-detection of a same vehicle. False detection identification considers the speed of a vehicle in relation to the traffic state, in which the system removes any unrealistically slow moving detection from counting (e.g. non-vehicle) at each timestamp. Finally, as soon as the vehicle passes into the counting zone and is successfully detected, the counter on the top of the video is incremented by one for the lane in which the vehicle was present, and the average speed of the vehicle is added to the list of reference speed which is iteratively used for false detection identification. By repeating above procedures for the entire duration of video samples, the total counting as well as lane-by-lane counting is provided.

As explained earlier, the detection performance can be characterized by the trade-off relationship between true detection and false detection of the trained template. For a counting application, a point on the curve should be chosen based on the location where the detector can maximize its true detection rate while keeping the false detection rate as low as possible. For a proposed framework where the metrics of performance is dependent not only on the detector used, but also tracking algorithms and the counting framework, the counting of vehicle can achieve its maximum performance by considering the spatial relationship among tracked vehicles during the tracking phase.

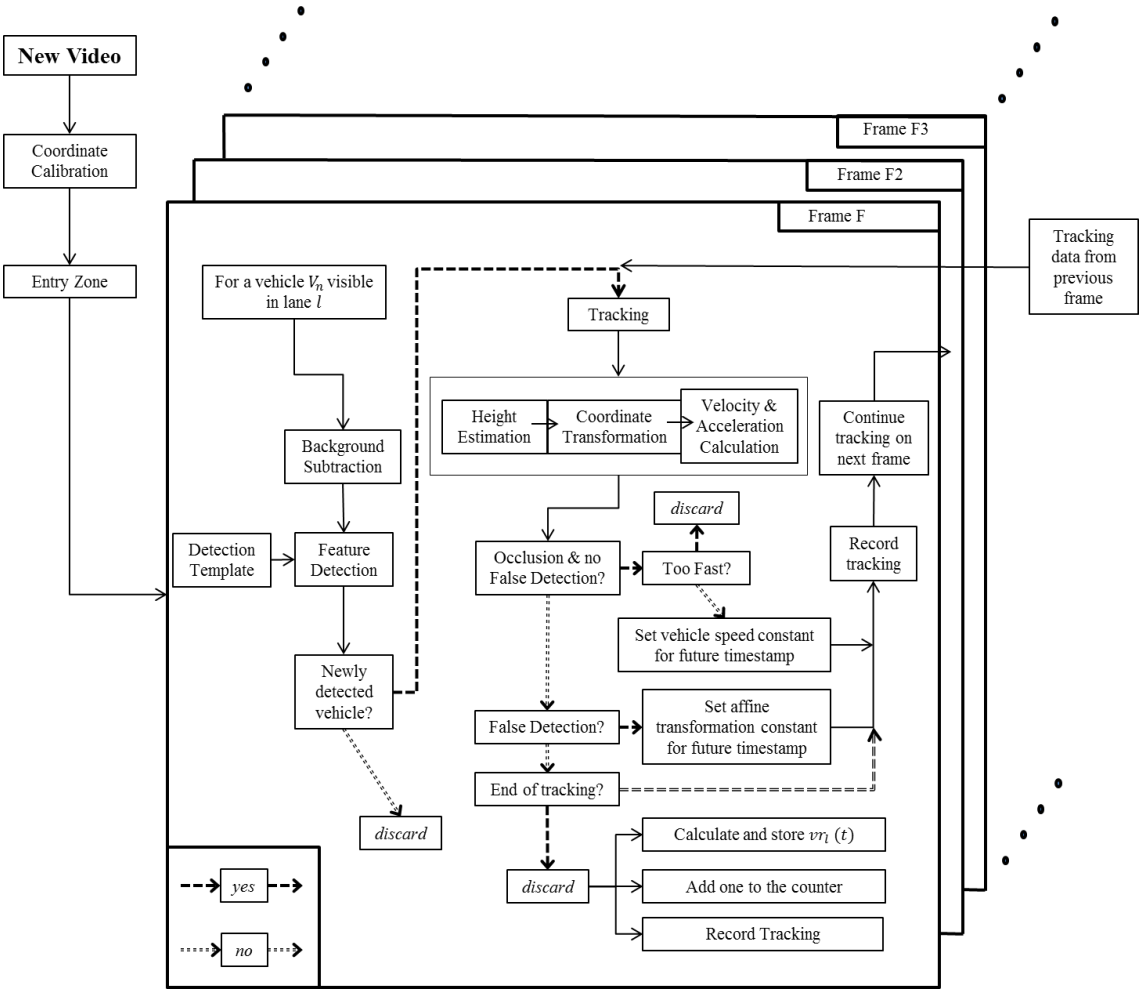


Figure 23: Framework of a Proposed Methodology

6.2 Modification to the Original Framework

In addition to the general framework explained in Chapter 4, several other functionalities are added exclusively for this vehicle counting application, including road plane tracking and reference speed calculation.

6.2.1 Tracking Based on 2-D Real Road Plane

Changes are made to the incremental tracking method such that this framework considers the tracking of vehicles based on the likelihood of change in direction in real-world road

plane coordinate (x_r, y_r) that are acquired through calibration, rather than the image plane (u_i,v_i) as originally proposed in (Ross et al. 2008). Considering vehicle on highway travel in a known direction or potential set of directions, this modification to the tracking algorithm facilitates the probability of each tracker moving from frame to frame in the direction of traffic flow much greater (thus, better tracking) rather than relying solely on the random probability described in the image coordinate which has no association to the traffic flow direction in the video. This modification also sets the threshold used for likelihood of tracking movement as a function of a velocity of moving traffic, thus allowing user to find the optimal tracking parameters effectively for a sample video.

6.2.2 Counting Zone and Reference Speed

When the user specifies the entry zone as in Figure 24 during the initial calibration process, the software automatically places the red boxes (called counting zone afterward) at the end of entry zone for each lane. For every vehicle that passes through this counting zone, the counter (the array of number on top of the frame in Figure 24) is incremented by one for that lane. The final total vehicle counts can be measured by the numbers displayed in the counter array next to the text "TOTAL:".

The counting zone also records the last successful average velocity of the vehicle upon entering. The velocity data are used for calculating the reference speed which is used for identification of false detections described in section 6.3. Because the beginning stage of vehicle tracking often can be unstable because of lack of stored subspace memory, the corresponding average vehicle speed takes time to stabilize. For example, speed data provided for a sample vehicle shown in Figure 17 at the end of chapter 4

initially have an average speed of 73.3 mph, but after about tenth of a second, stabilize to about 66.1mph for all subsequent tracking. Thus, the average speed of vehicle crossing the counting zone is counted toward the reference speed of that lane only if the tracking duration of such vehicle exceeds the threshold (in this thesis, frame rates equivalent to approximately one-half second are used). This approach is implemented to reduce the likelihood of introducing an unstable velocity data into reference speed pool.

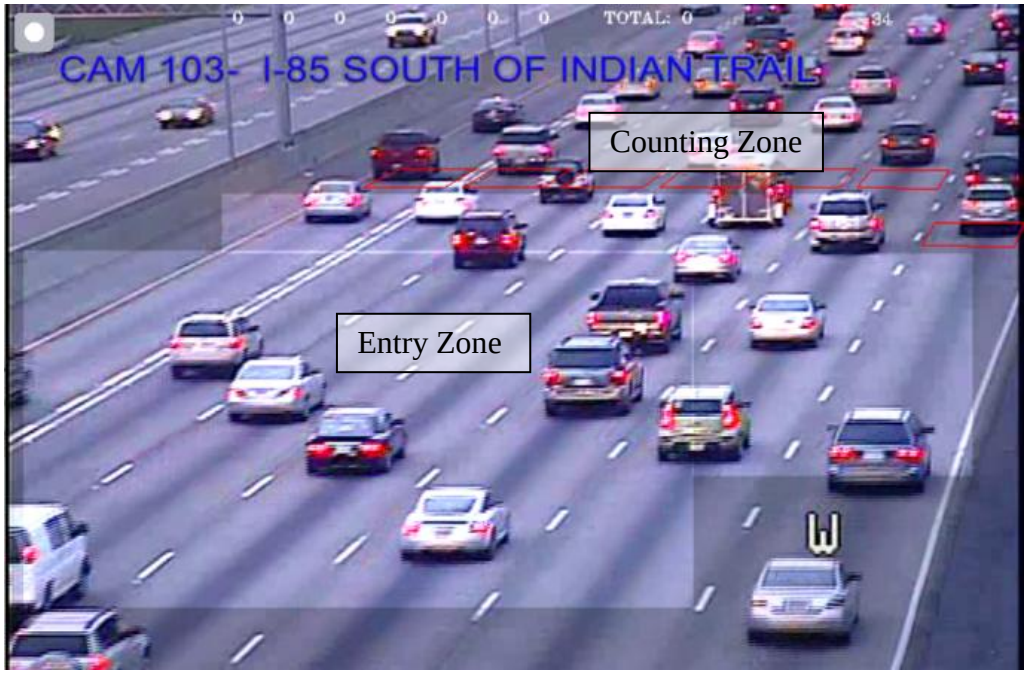


Figure 24: Entry Zone and Counting Zone

6.3 False Detection Identification

In this thesis, the proposed framework relies on the relationship that exists between traffic density and speed to identify the falsely detected element that are too slow or too fast compared to the traffic flow. In general, traffic density is inversely proportional to the traffic speed; if the vehicle speed is high, traffic density is expected to be very low, and vice versa. However, this relationship generally applies to the data that are averaged,

usually observed for fifteen minutes, one hour, or longer period. At any given time *t,* the *instantaneous* characteristics of traffic flow may behave differently compared to the averaged data with greater variability. Also, corresponding traffic flow can be described relative to the surrounding flow characteristics such that large variations for vehicle-to-vehicle speed can rarely be seen. Although there is some exception to this rule, as seen in Guin et al. (2008) which reports large speed disparities between High-Occupancy Toll (HOT) lane and adjacent general purpose lane at same time of a day, but the lane-by-lane speed characteristics can still be modeled by the speed in the same lane independent of other lanes. As such, the system focuses on the minimum observed speed at any given time because slow moving object are typically a candidate of a non-vehicle having low speed compared to the traffic flow.

The proposed framework takes advantage of this phenomenon for identifying the non-vehicle elements that are falsely detected in the detection phase. Assuming that there are no other in-motion objects other than the vehicles in highway, the falsely detected objects are usually a part of road, median, shoulder, or poles that appear still in the sequence of frame. This assumption works because in the highway traffic scene, there are no other in-motion objects (except the casted shadow of a vehicle) that move in the same direction as the traffic flow. During the tracking phase, these objects often stay still because of lack of apparent movement in non-vehicle regions, resulting in significant decrease in its instantaneous velocity. Thus, the high-pass filter with an adaptive threshold v_{min} can be constructed to exclude these outliers relative to the traffic speed.

As shown in Figure 25, the speed of non-vehicles theoretically falls under the Area A in the density vs. speed plot. On the other hand, the filtering cannot operate

unless there are data with which comparison can be made. Since the variability of speed vs. density data varies significantly in a microscopic sense, such generalized relationship cannot necessarily serve as the reference point, nor can be assumed to be linearly correlated. Also as discussed earlier, each individual lane in the highway traffic behaves differently from one another in terms of their speeds (Guin et al. 2008).

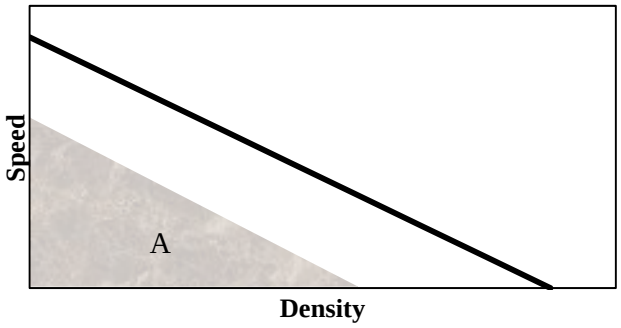


Figure 25: Typical Linear Relationship between Traffic Density and Speed; the Location of Area A

Thus, the reference data that filtering relies on should be a representation of a localized traffic behavior at timestamp *t* for each individual lane. Consequently, in this framework, the speed of last five vehicles is stored in the system per lanes at an instance when the tracking is terminated for each object (in other words, when the vehicle enters the counting zone). The average value of the stored velocity ($vr_{F,l}$) provide an estimated reference speed per lane at time *t*, on top of the constant threshold of 20 mph providing more confidence in case of an outliers. In the start of video, the N value in Equation 6.1 increments by one for every vehicle added to the reference velocity until the counts reach five, at which point the further addition to the reference speed calculation discards the oldest velocity data. If speed of any tracked object (v_F) falls below the reference speed

$vr_{F,l}$ of that lane, then such detection is likely a false detection. In Figure 26, tracking region 97, 99, and 108 has been removed because the speed of such tracking region fell below the reference speed.

1) $v_F < vr_{F,l} = \frac{\sum_{k=1}^{N} v_{k,t_{end},l}}{N} - v_{th}$ (Eq. 6.1)

In addition to removing detected objects based on relative speed, the scale value is also considered for criteria of removal. The basic underlying idea is similar to the aforementioned local speed relationship, where if one detected object behave differently than the rest of object pools, such object is likely to be falsely detected object. Likewise, if the scale value S_F(e.g. an area covered in image coordinate for each region R_T) of one object becomes too small because of a lack of discriminative features inside tracked region R_T compared to the averaged scale of all tracking objects (S_{avg}), then the object is more likely to be a non-vehicle. Here, 25% of S_{avg} is set as criteria of removal. In Figure 26, the tracking region 92 has been removed because the area inside the pixel region fell below the quarter of average scale threshold.

2) $S_F > \frac{S_{avg}}{4}$ (Eq. 6.2)

As discussed in Chapter 4, the instantaneous and even the averaged speed data are greatly affected by the varying frame rate. Thus, there is a probability of a vehicles' detected region being incorrectly identified as a non-vehicle through this filtering operation. To minimize this problem, the tracking of each object contains the default constructor where for every time the speed fall within the filter range as specified by Equation 6.1 or 6.2, its value is incremented by one. If this value exceeds or equals the

pre-determined threshold (in this framework, a constant threshold of two is used), then such tracking can finally be considered as non-vehicle.

If the detected object is simply *removed* from the processing every time the false detection is recognized, the systems typically detect the same exact detected region D_T repeatedly because of similarities in features that appear in the image. This additional processing is undesirable as it iteratively introduce the chance of additional errors to the count results. Thus in this framework, the systems will keep the tracking region R_T upon the recognition of false detection and continue to track the falsely detected object resetting its affine transformation parameters used for the tracking to zero. This way, the algorithms for newly detected vehicle provided in Chapter 4 continuously prevents the new detection of object surrounding that region, and the counting of this object simply does not instantiate once it reaches the counting zone. To indicate that the objects are removed from counting, such objects will be marked by green rectangle, as shown in Figure 26.

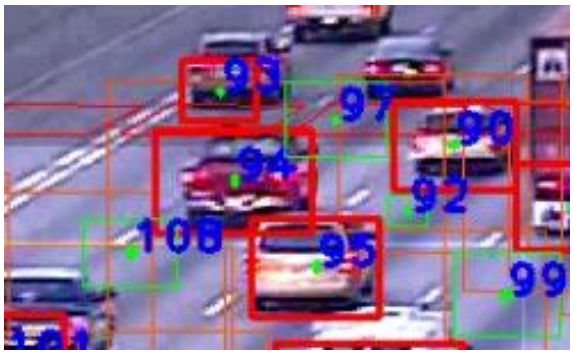


Figure 26: Example of Object Removal Process

6.4 Occlusion Identification

The nature of occlusion handling is much more complicated than the identification of false detections because: 1) high variance in speed fluctuation often occurs at the instances when the clear view of a vehicle is hindered, 2) different types of occlusion must be handled differently, as described more in details in section 6.4.1, and 3) the relationship in relative speed between the occluding and occluded vehicle makes identification challenging. As such, pattern recognition regarding different types of occlusion scenarios as well as its relations to the general traffic movement can be hard to generalize. In a microscopic sense, a clear view of a single vehicle moving at constant speed can be hindered by wide variety of velocity ranges in the adjacent lane (anywhere from 0 mph to well above design speed) depending on congestion level on a per-lane basis. In this thesis, the occlusion handling is approached by the modification to the tracking algorithms combined with acceleration data.

The use of tracking algorithm presented in Ross et al. (2008) offers a unique solution to deal with the phenomenon of vehicle image *disappearing* from the scene when such vehicle is completely occluded by the adjacent larger vehicle. The accuracy and efficiency of this tracking algorithm largely depends on the presence of clear vehicle image that is initiated by the detection phase. The algorithm continuously recaptures and updates the view of this entire vehicle element throughout the vehicle movement and store the spatial data into the temporary subspace, while managing to retain the accuracy even when the vehicle is rotated or skewed as a result of steady movement such as lane changing. As soon as the vehicle's clear view is disrupted by the occlusion, the tracking that is based on the past stored subspace becomes uncertain because the abrupt change in

the view does not reflect the stored spatial view. This essentially causes the tracked region of occluded vehicle “wobbles" around the region where the object was last tracked successfully, continuously diminishing in the scale value as being pushed forcibly by the occluding vehicle. The proposed method will recognize this wobble movement as reflected by the abrupt change in the scaling of tracked area (S_F) compare to that of previous frame (S_{F-1}). If the ratio of two scaling value are so great that change in scale exceeds the threshold value while managing to stay above a 25% of average scale values, then the object is likely to be occluded.

1) $\frac{S_F}{S_{F-1}} < S_{th}$ (Eq. 6.3)

2) $S_F > \frac{S_{avg}}{4}$ (Eq. 6.4)

Similar to the identification of false detection framework, the occlusion handling framework also stands by the conservative setting where for each possible occlusion recognized, the constructor of each tracking is incremented by one. If the value equals or exceeds the threshold (in this research, the constant value of 2 is used), then the system finally recognizes the vehicle has been occluded.

Preliminary empirical study indicates that the labels filtered out by the above occlusion handling does segregate the occluded data from the rest fairly well, but also contains extraneous data that could be false detection (non-vehicle). These false detection that are misclassified as the occluded vehicle typically reduce in scale value substantially in traffic movement by the wobble movement described above by being pushed up (or down based on traffic movement) by the occluding vehicle that comes after it, having extremely high, nonrealistic fluctuation in average speed. The true occluded vehicles

typically reduce in scaling factor gradually and steadily while retaining a relatively constant average speed (thus, having low average acceleration $a_{avg,F}$). Thus, additional processing was added to sort out the true occlusion by reinforcing the algorithm with acceleration data.

3) $a_{avg,F} > a_{th}$ (Eq. 6.5)

where: $a_{avg,F}$ = averaged acceleration of a vehicle at a particular frame

a_{th} = threshold acceleration = $106*10^3$ miles/h^2

(empirically chosen)

As explained in Chapter 4, the acceleration data are unreliable without additional smoothing due to varying frame rates. In this application, however, these data serve effectively as a mean to compare the result produced from the first occlusion handling based on the scale factor. This low-pass filter (i.e. high acceleration is an indicator of a non-vehicle) compares the current vehicle acceleration to the threshold. If the given value exceeds the threshold more than twice, the label is officially registered as an occluded vehicle. The detected occlusion label will be marked by yellow enclosing rectangle as shown in Figure 27.

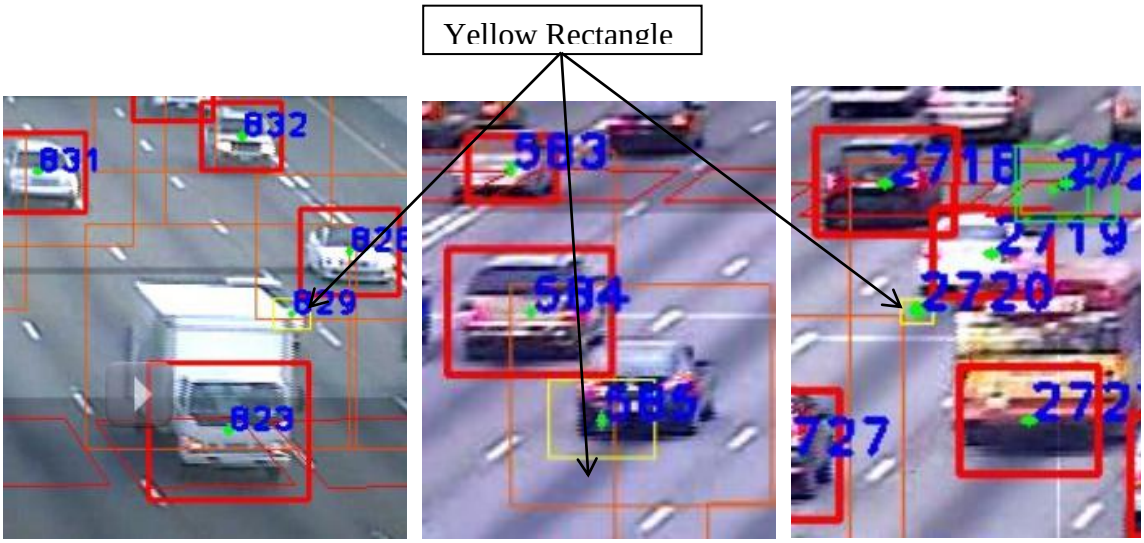


Figure 27: Occlusion Detection (marked by yellow rectangle): correct detection (left), incorrect detection of vehicle (middle), incorrect detection of non-vehicle (right)

6.4.1 Occlusion and Counting Data

In total, all occlusion scenarios observed in the highway traffic can be described by a combination of Clear state (denoted as C) and Occluded state (denoted as O). They include 1) when a vehicle's view is clear at first, then become occluded (C-O, Figure28), 2) when a vehicle is initially occluded, but firstly appear once the occlusion is cleared (O-C, Figure 29), 3) when a vehicle's view is clear for the entirety of its movement (C), and 4) when a vehicle is completely occluded for the entirety of its movement (O, Figure 30). In this experiment, the occlusion type 4 is beyond the scope of this research as such occlusion is physically impossible to be recognized in any monocular vision-based detection-tracking systems.

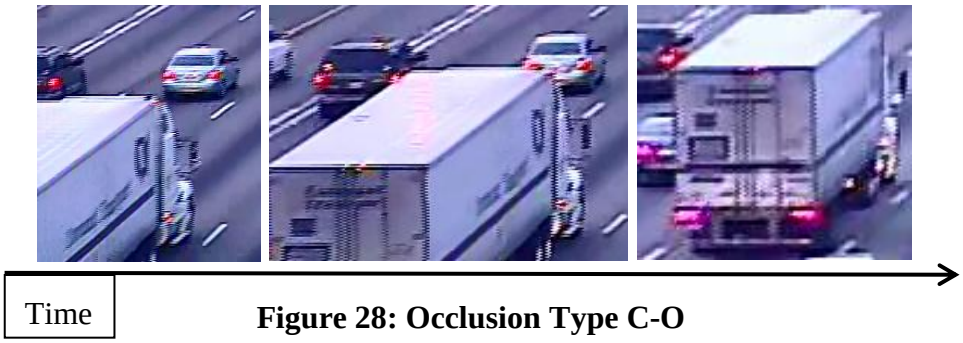


Figure 28: Occlusion Type C-O

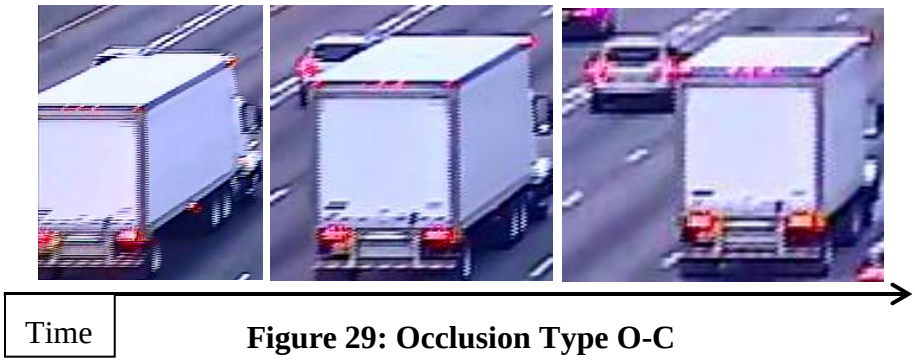


Figure 29: Occlusion Type O-C

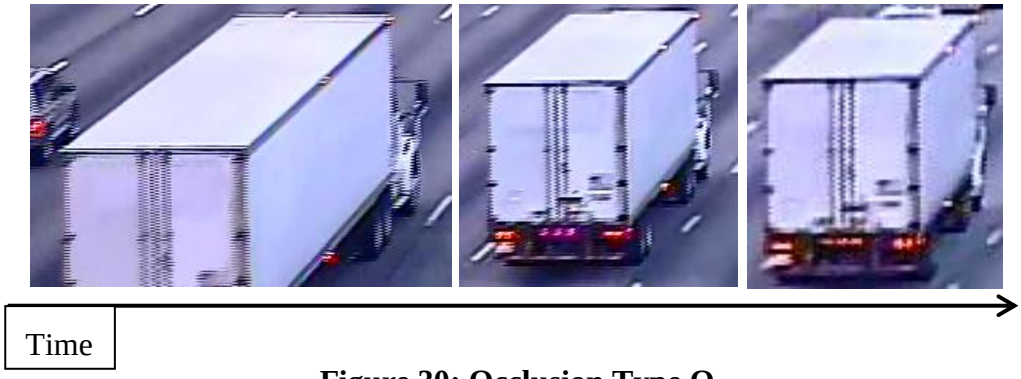


Figure 30: Occlusion Type O

For occlusion type O-C and C, occlusion handling techniques are unnecessary as the correct vehicle count will be added to the counter as soon as the detected vehicles arrive at the counting zone. However, C-O occlusion, and combination of C-O and O-C occlusion can be problematic because of undercounting or overcounting. During C-O occlusion, if the detected vehicle is to be tracked with the same configuration as before, it often results in undercounting because of the wobble movement increasingly affect the

tracking accuracy as reflected in the sudden increase in the change of scaling, thus recognizing it as the falsely detected object through the identification of false detection described in section 6.3. If the object removal somehow failed and detected object are continuously tracked even after the occlusion has been cleared (this occlusion type can be described as C-O-C: occlusion type C-O, then occlusion type O-C; refer to Figure 31), overcounting can be observed to as the same vehicle are detected twice, both counted toward the counter as distinct objects. In other combinatory case (e.g. O-C-O, occlusion type O-C, then occlusion type C-O; Figure 32), occlusion handling is taken care during the latter occlusion type C-O as no detection are triggered during the first occlusion type O-C.

Figure 31: Occlusion Type C-O-C

Figure 32: Occlusion Type O-C-O

The general conservation of mass theory in highway traffic also motivates the use of proposed occlusion handling techniques, which states that every object (e.g. vehicle) coming into the system must leave the systems. Even if the occlusion can be detected and its occluded vehicle's movement record is removed from the counting data, that data is lost forever while the system recognizes that there *"was"* a vehicle present in several past frames. This data must be retrieved by the systems for the occlusion handing to be deemed effective.

For occlusion type C-O, the proposed framework retain the tracking of a vehicle for the future subsequent frames in the video by assuming that the y-direction (i.e. parallel to the movement of traffic) speed is constant and x-direction (i.e. perpendicular to the traffic movement) speed remains the same the moment when the tracking is recognized to be occluded. This also allows an opportunity for any vehicle that are classified as occluded an ability to retrieve its data and count toward the counter even if its view is not occluded to begin with (e.g. falsely detected occlusion, Figure 27). In case of occlusion C-O-C, the tracking of the existing vehicle terminates as soon as the first occlusion is detected, allowing the subsequent re-detection to take over the first detected

object. This approach essentially gives two distinct vehicle IDs for same one vehicle, but the counting of the overall traffic will only count once as the first detected ID will be removed. For used samples where the field of view is relatively narrow, these subsequent occlusions typically occur when the speed of occluded vehicle is comparatively low. Therefore, the initial tracking data is discarded only for vehicle having speed less than minimum specified speed (In this thesis, 30 mph threshold is consistently used) to differentiate between C-O and C-O-C occlusion. It is important to note that the systems does not deliberately identify the types of occlusion, but rather work with scaling value, acceleration, and averaged vehicle speed data directly.

CHAPTER 7

TESING, RESULTS AND DISCUSSIONS

7.1 Training Detector

A total of 1801 and 2181 positive samples (for rear view and front view, respectively) were trained with 4258 negative samples to form two detectors for two different views. Figure 33 shows some sample training images. In this thesis, 4258 non-vehicle images are consistently used for all experiments to keep the consistency in the training process.

Rear View

Front View

Figure 33: Sample Images Used for Training

7.2 Description of Testing Samples

The sample traffic videos are extracted from CCTV cameras throughout the north-east corridor of Interstate-85 in the state of Georgia. Four 15 minutes sample videos were

chosen from several different locations during the daytime. In this thesis, the author manually counted traffic volume for sample 1, 2 and 4, and undergraduate students hired on hourly basis manually counted volumes in sample 3. Manual counting data are assumed to be the ground truth although some error may exist. Table 3 shows the detailed description of environmental variables for each video samples. Figure 34 displays the field of view for all samples, but the exact map locations of stationed camera can be found at Georgia NaviGAtor website.

Table 3: Description of Environmental Variables for All Test Samples

Samples	Camera #	Traffic Direction	Object Size	Illumination	Frame Rate	Congestion Level	Camera angles with respect to the road	
							θ (in degree)	φ (in degree)
1	#103	Away from Camera	Med - Big	Mildly dark	Relatively Stable	Free flow, then Heavily Congested	6.83	5.73
2	#212	Away from Camera	Small	Cloudy	Unstable	Free Flow	13.4	9.28
3	#101	Toward Camera	Med-Big	Sunny	Relatively Stable	Congested all round	7.45	6.15
4	#101	Toward Camera	Small	Foggy	Relatively Stable	Mildly Congested	5.71	8.24

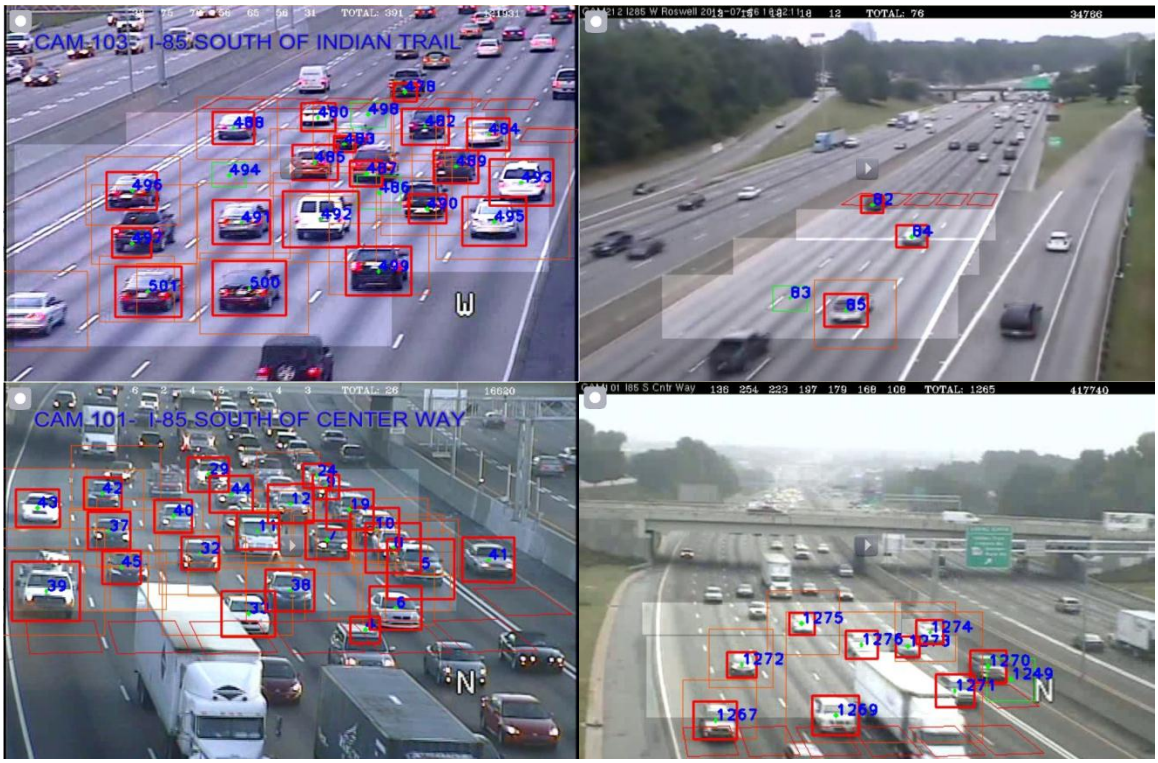


Figure 34: Screenshots of Interface for All Samples: Sample 1 (Top Left), Sample 2 (Top Right), Sample 3 (Bottom Left), Sample 4 (Bottom Right). Note: Orange Rectangle Represents the Results of Background Subtraction.

7.3 Performance Metrics

The following four testing parameters evaluate the performance of system for each video.

1) Correct Counting Rate (CCR)$= \frac{\text{Counts that are vehicle}}{\text{Manual counts}}$

2) False Counting Rate (FCR)$= \frac{\text{Counts that are non-vehicle}}{\text{Total counts of the automated system}}$

3) Removal Rate RR (RR)$= \frac{\text{Total number of falsely detect vehicle removed from counting}}{\text{Total number of falsely detected object}}$

4) Occlusion Handling Rate (OHR) $= \frac{\text{Number of vehicles correctly added by detected occlusion}}{\text{Total detected occlusion}}$

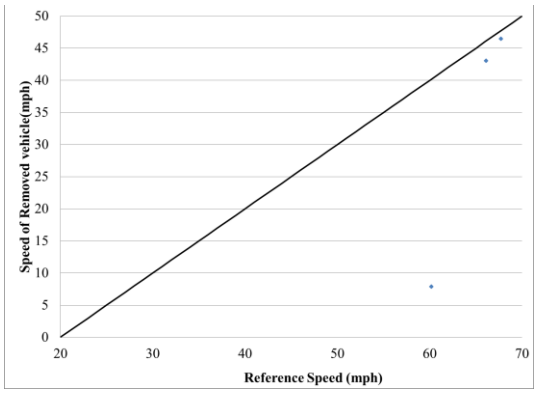
The results from each video samples are also compared among themselves in terms of observed environmental variables such as illumination variance, detector power, congestion level, and the difference in camera angles (θ, φ, horizontal and vertical angles, respectively) with respect to the road in which the classifying template is trained.

7.4 Verifying Reference Speed Threshold

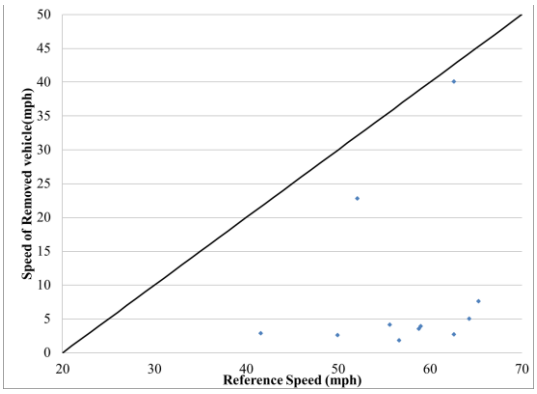
The preliminary experiments indicate that the reference speed (i.e. last five average speed stored per lane, see section 6.2.2) does indeed provide a good estimation of where the range of vehicle speed should fall. In Appendix C, the plots for all used sample videos are provided to demonstrate the correlation between calculated reference speed and observed minimum speed per lane at timestamp t. As the result, the calculated reference speed often seems to closely fit to the minimum observed vehicle speed. Some outliers in the minimum speed data are present because; 1) there is some degree of randomness in the probability-based tracking algorithm, or 2) the label is classified as a false detection only once at that particular timestamp t (i.e. need to pass minimum of two false detection filter to truly be recognized as a false detection). Figure 35 provides the distribution of speed of removed object versus the reference speed upon removal for sample 1 at minimum neighbor of two. As shown, speed of removed object always falls below the threshold reference speed (e.g. 20mph). This figure also indicates that the simple constant threshold filter with no relations to the traffic condition would not work, as the velocity of falsely detected element that are based on observed minimum speed is widely distributed in the region below the threshold line. The results justifies the use of v_{th} of 20 mph can

effectively remove good amount of falsely detected elements given an approximate 90% RR shown in Appendix B, while keeping most of true detected vehicle.

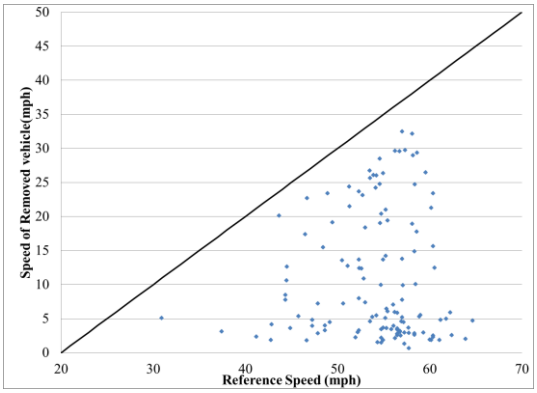
Lane 0



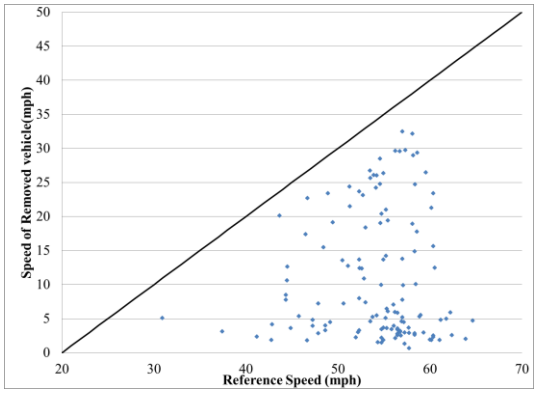
Lane 1



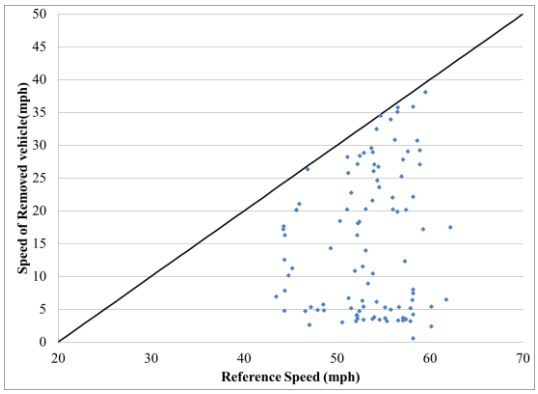
Lane 2



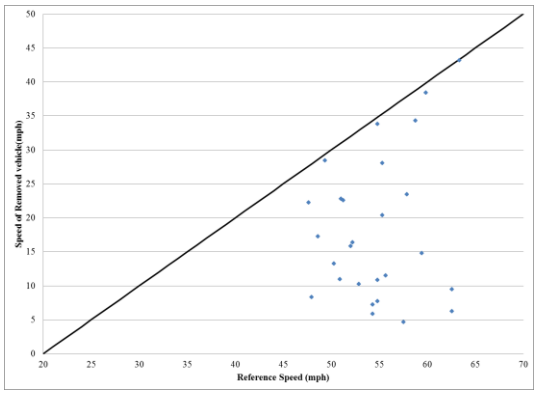
Lane 3



Lane 4



Lane 5



Lane 6

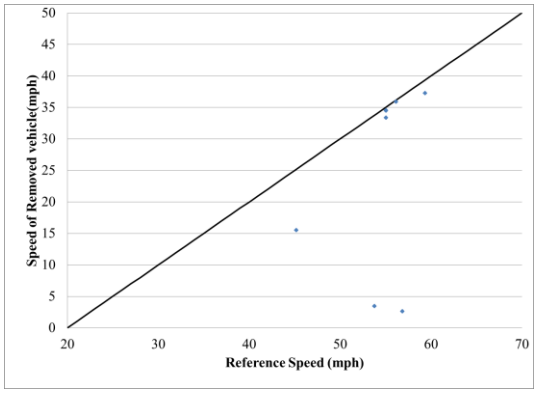


Figure 35: Reference Speed vs. Speed of Removed Vehicle (From Equation 6.1)

7.5 Results of Performance Metrics

Table 4 below provides the cumulative results for each video at minimum neighbor of one to five. As expected, the total counts are observed to be increasing as minimum neighbor threshold decrease, marking very high correct counting rate The results also indicates that the system can cope with variety of environmental variables (e.g. difference in illuminations, congestion level, object size, etc.) with respectably high correct counting rate for all video samples. Note that some occlusion handling rate and removal rate is extremely low because of negligible sample size. Especially for removal rate, the true performance of a low pass filter is represented for videos with minimum neighbor of one where there is abundance of false detection candidates. Nevertheless, the removal rate of each vehicles based on the used high-pass filter can be demonstrated to be quite effective while retaining the false count rate to be reasonably minimum. For sample 1 when minimum neighbor of one, as many as 985 detections were removed by the low pass filter, of which only 18 of them were actual vehicles (Table 4). On the other hand, for sample containing extremely unstable frame rate (e.g. sample 2), the removal rate is shown to be low as removal process excessively remove true vehicle containing high

speed variation. Although relatively less effective, occlusion handling technique used in this research does seem to contribute to the correct counting rate as indicated in occlusion handling rate value. Some vehicle was falsely detected that the vehicle is occluded; however, this does not jeopardize the overall counting results as the vehicle count will be added to the counter regardless. The occlusion handling also does not seem to trigger excessive false alarms in case of when there are not many occlusions to begin with (i.e. the occlusion handling was not triggered for sample 4 in most cases). Appendix B provides raw counting data for all samples.

Table 4: Overall Result of Testing Parameters for All Samples

Sample 1						
Minimum Neighbor		1	2	3	4	5
Correct Counting Rate (CCR)	n (Manual Count)	3046	3046	3046	3046	3046
	(%)	**95.2**	92.2	87.6	83.6	78.6
False Count Rate (FCR)	n (Total Detection)	3051	2875	2689	2557	2400
	(%)	4.9	2.3	0.8	0.4	0.3
Removal Rate (RR)	n (Total Removed)	985	749	250	137	88
	(%)	98.2	94.7	95.2	91.2	84.1
Occlusion Handling Rate (OHR)	n (Total Detected Occlusion)	39	29	12	17	12
	(%)	79.5	69	58.3	82.4	83.3
Sample 2						
Minimum Neighbor		1	2	3	4	5
Correct Counting Rate (CCR)	n (Manual Count)	1550	1550	1550	1550	1550
	(%)	**93.2**	91.5	88	85	81.4
False Count Rate (FCR)	n (Total Detection)	1526	1455	1381	1325	1265
	(%)	5.3	2.5	0.7	0.5	0.3
Removal Rate (RR)	n (Total Removed)	341	133	77	49	53
	(%)	90	72.9	55.8	40.8	41.5
Occlusion Handling Rate (OHR)	n (Total Detected Occlusion)	10	4	4	5	1
	(%)	90.9	100	100	100	100
Sample 3						
Minimum Neighbor		1	2	3	4	5
Correct Counting Rate (CCR)	n (Manual Count)	2667	2667	2667	2667	2667
	(%)	**97.7**	96.4	95.6	94.1	93.5
False Count Rate (FCR)	n (Total Detection)	2660	2612	2562	2515	2498
	(%)	2.1	1.5	0.5	0.2	0.2
Removal Rate (RR)	n (Total Removed)	34	22	20	14	15
	(%)	91.2	59.1	75	57.1	66.7
Occlusion Handling Rate (OHR)	n (Total Detected Occlusion)	4	3	3	4	5
	(%)	57.1	100	100	100	83.3
Sample 4						
Minimum Neighbor		1	2	3	4	5
Correct Counting Rate (CCR)	n (Manual Count)	2669	2669	2669	2669	2669
	(%)	**98.8**	98.3	97.8	97.2	95.7
False Count Rate (FCR)	n (Total Detection)	2691	2649	2629	2606	2559
	(%)	2	0.9	0.7	0.5	0.2
Removal Rate (RR)	n (Total Removed)	54	23	11	5	11
	(%)	87	91.3	72.7	60	54.5
Occlusion Handling Rate (OHR)	n (Total Detected Occlusion)	0	0	0	0	1
	(%)	NA	NA	NA	NA	100

As shown in Appendix C, the calculated reference speed resembles the shifted version of moving average speed, which is reasonable because the speed calculation is conducted at the end of tracking. Even then, the performance of the proposed framework seems promising.

The system however does not seems to work well for all camera angles. Figure 36 shows the variability of lane-by-lane counting rate of all samples. Especially for Sample 1 at lane 0, the correct counting rate only marked 64.9 % while the rest of lanes demonstrated significant better correct counting rate. While the lack of enough training data from the specific view of lane 0 may certainly affected the outcome, the low correct counting was also influenced by 1) the faster vehicle passing through HOT lane such that the vehicle remain in entry zone much less time compared to the rest of the lanes, and 2) the relatively smaller field of view for lane 0 based on the positioning of PTZ camera (see Figure 34). This part of analysis conversely justifies the approach considered in the proposed method where the detection rate in the detection-tracking framework reaches maximum as vehicles remain in the detection zone (e.g. entry zone) longer, and insinuate that the positioning of PTZ camera in this framework plays an important role.

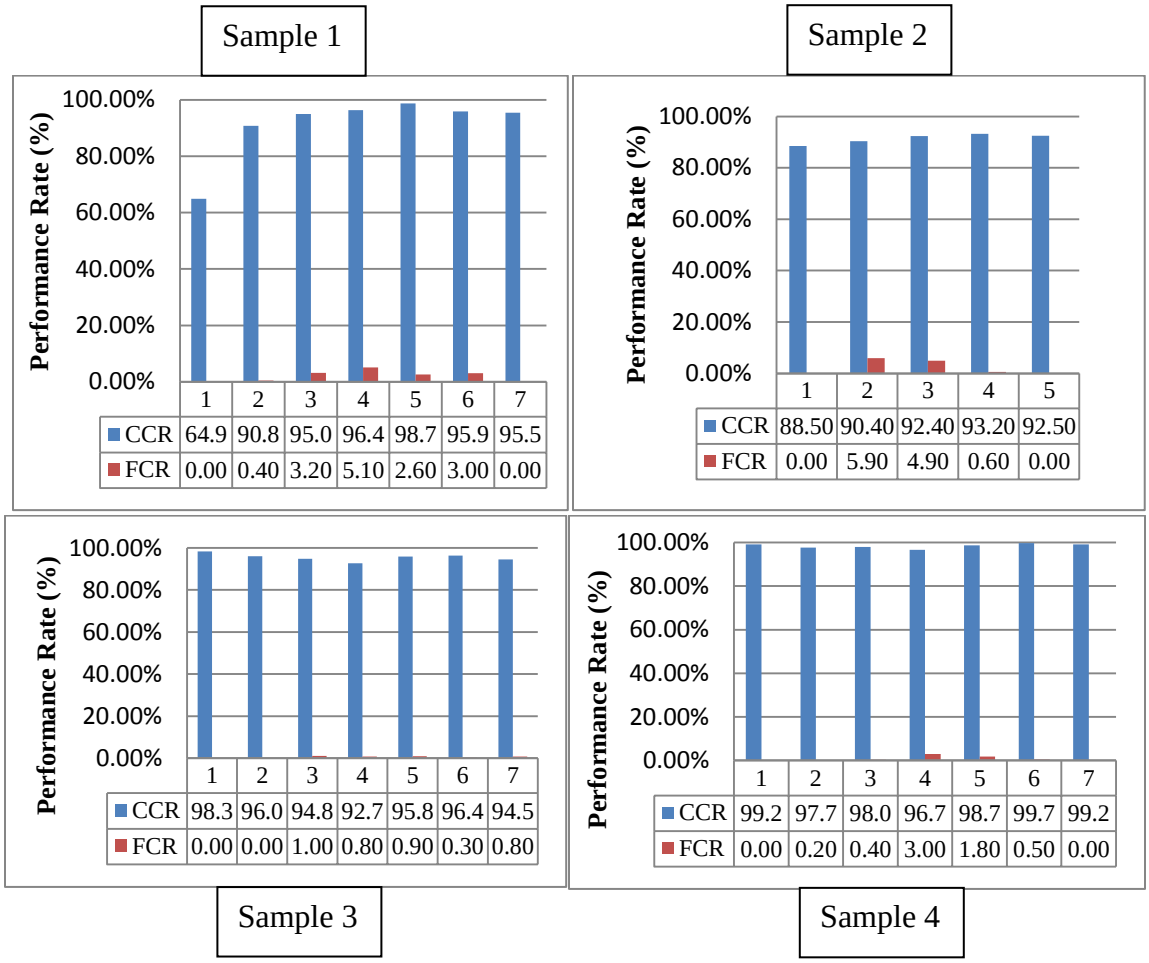


Figure 36: Lane-by-Lane Correct Counting Rate (CCR) and False Count Rate (FCR) for All Samples (Minimum Neighbor=2)

7.6 Error Analysis and Limitation of a Methodology

The nature of probabilistic-based tracking algorithm used in this proposed framework poses difficulty pinpointing out the contributing factor to the overall counting error. Still, there are several recognizable errors that are rather evident.

7.6.1 Speed Variability in Various Congestion Level

Throughout this experiment, the operating characteristics of each vehicle are assumed to be uniform and correspond to general traffic flow. This is not the case in a real-world scenario when the excessively aggressive or overly timid drivers are present in small numbers. In the sample videos used, this seemed to affect counts more significantly when the traffic was free-flowing (e.g. sample 2 and 4). The false count rate in those relatively less congested sample videos may have increased partly because the drivers gain more freedom in their driving pattern due to greater sight distance between vehicle, making each drivers comfortably drive on their preferred speed rather than adhering to the general traffic flow as in a congested traffic. Figures from Appendix C also justify this claim, indicating higher correlation coefficient value, *r* for the relationship between minimum speed and reference speed for sample 1 and 3. Nevertheless, this phenomenon poses a challenge in the used high-pass filter operation where extremely timid drivers as well as many heavy-truck drivers may tend to travel much slower than the observed traffic flow. The result from Appendix B reflects the error caused by timid drivers. For example for sample 4, one vehicle in lane 5 was consistently removed for all minimum neighbor values tested because this vehicle traveled way much slower compare to the rest of traffic state.

Large speed differential between HOT lane (lane0) and adjacent general purpose lane (lane 1) as reported by Guin et al. (2008) also poses additional problem for a weaving vehicle based on the difference in congestion level. Figure 37 shows an example of a vehicle (in sample 3) weaving into HOT lane from adjacent general purpose lane when congested. As soon as the system recognized this vehicle belonging to lane 0, the

filtering operation discussed in section 6.3 immediately removed this vehicle because of the large difference in the calculated reference speed stored for each lane. As shown in Figure 38, the difference in calculated reference speed between lane 0 and lane 1 during this weaving is about 35 mph, which is greather than the used 20 mph threshold, thus this particular vehicle was accidently removed by the filter. During the 15-minute sample, this type of an error happened only once, but was consistantly observed for all of minimum neighbor values tested.

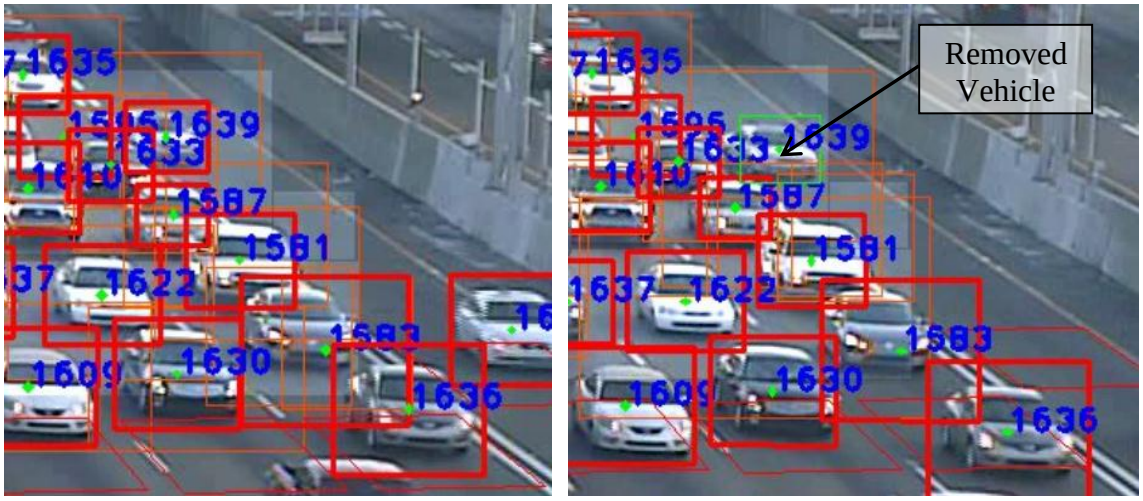


Figure 37: Count Removal during Weaving into HOT Lane

Sample 3

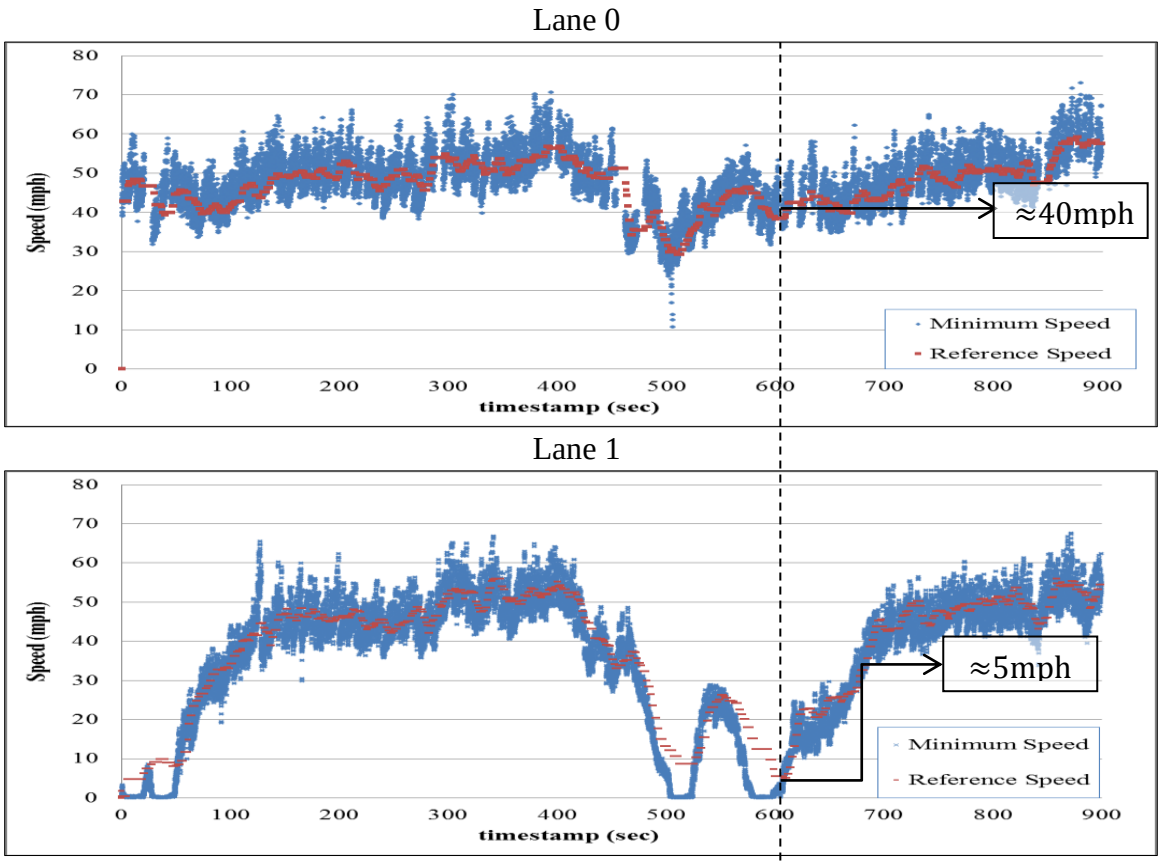


Figure 38: The Speed Differential during the Observed Weaving (Same Figures Presented in Appendix C)

7.6.2 Propagation of Errors within Vehicle Platoons

It is common knowledge in traffic engineering that moving traffic tends to form platoons of vehicles, where the leading vehicle of each platoons may constrain the speeds of following cars (Roess et al. 2004). This traffic characteristic ultimately has a great impact on the used filtering operation where the reliability of successful tracking removal largely depends on each successive vehicle speeds that are being tracked. Thus, if series of vehicles were not successfully detected by the systems, the stored reference speed fails to

cope with the abrupt change in the speed differentials even with the used 20 mph threshold. For example in sample 1 testing, the total number of falsely removed vehicles by the filters topped 23 vehicles for lane 0 when minimum neighbor of two was used (see Appendix B), compared to only 8 removed vehicle when minimum neighbor of one was tested. This undesirable result was caused by the leading vehicle's velocity falling below the minimum reference speed threshold, making all of the vehicles that follow the leading vehicles falling below minimum as well (Figure 39). Probabilistic nature of tracking algorithm also contributes to the cause as the error only can be seen for minimum neighbor of two (and not for all other values).

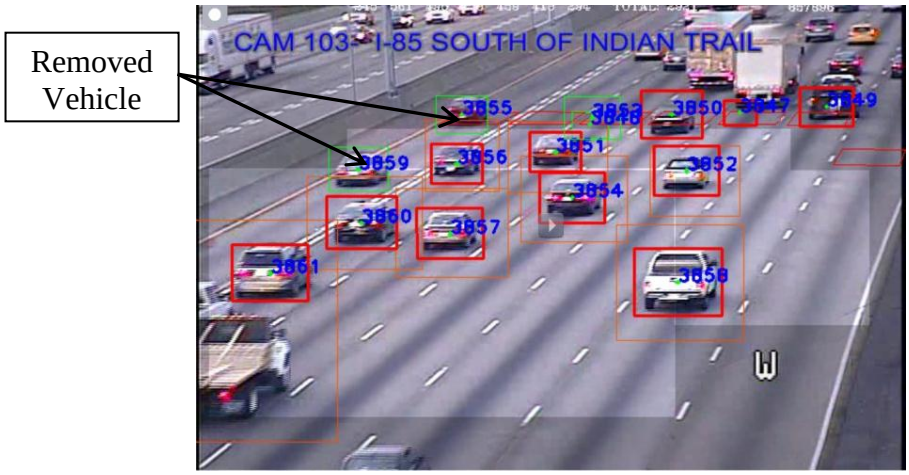


Figure 39: Repeated Removal of True Vehicles (lane 0)

7.6.3 Multiple Detection of a Single Vehicle

In this research, the double detection of a single vehicle (Figure 40) was beyond the scope of this research and was not exclusively dealt with throughout the course of the experiment. This resulted in a substantial increase in the false count rate especially when the minimum neighbor values for the detection phase were relaxed. As shown in Appendix B, the double detection counts as well as the false counts steadily increased as

the minimum neighbor value was reduced for all samples. This result can be misleading, because the proposed high-pass filter exponentially removes increasing number of false counts very effectively with removal rate of 90% or greater in most cases. However, the remaining double detected counts still pose an undesirable increase in false count rate, which cannot be addressed with the proposed system at this moment.

7.6.4 Spillover Effect

Dealing with large sized vehicles can be a difficult task for the reason mentioned in the previous section. Not only trucks are the biggest contributor to the additional false detection, but they also pose an additional problem called “spillover” effect firstly presented in Chapter 3. If the detection fails to represent the whole view of an object but rather capture only part of it, the counting data is often misplaced into the adjacent lanes, as shown in Figure 40. In the case of large sized truck detection, because the trained template did identify the exclusive features present in the truck image, the detector often times capture only part of its view. Conservation of mass still stands and the overall correct counting rate of the video sample will remain the same, but the lane-by-lane counting data may not represent the true traffic counts for that lane. In the test samples, sample 3 greatly suffered from this phenomenon as bigger object sizes (by pixels) are more prone to double and triple detections of large trucks.

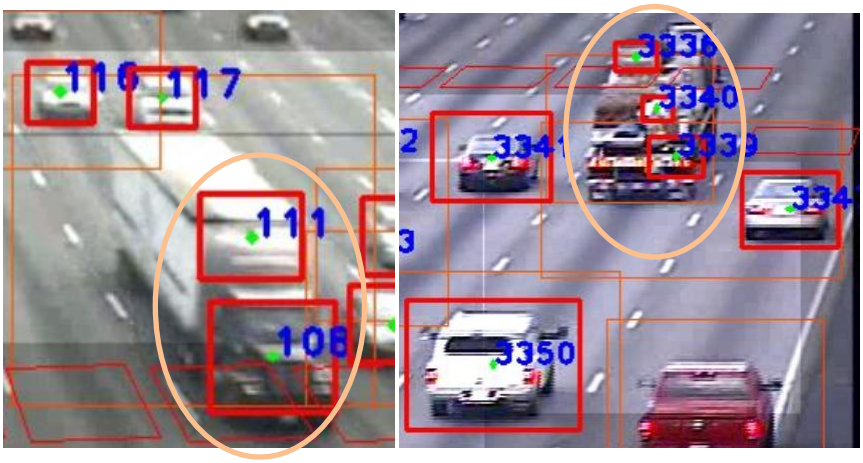


Figure 40: Multiple Detections of a Single Vehicle

This problem also could arise for regular light-duty vehicles. The quality of tracking largely depends on the initialized label produced by the automated feature-based detection. If the label was displaced or allocated into a location such that it does not capture the vehicle as a whole, the tracking is more likely to fail due to the incremental tracking algorithm because only the data stored in the subspace is continuously updated. Simply put, the tracking will continue based on the image captured by the initialization phase. Additionally, tracking of such part-filled vehicle object can affect the reliable counting data especially in the proposed framework where the scaling factor of this tracker is weighted for the filtering operation for removal. For instance, one vehicle displayed in the video (Figure 41), starts its travel from lane 3 and changes into lane 2 little before the tracking is terminated. If only part of that vehicle is detected (and not the entire view of the vehicle), as shown in Figure 41, the trajectory of the vehicle does not accurately corresponds to the ground truth movement of the vehicle. Instead, the output trajectory is the horizontally (or vertically) shifted-version of the ground truth.

Despite such unfortunate result in the translational movement caused by the misplaced initialization, the same figure also indicates that the occurrence of weaving is still apparent from the output data. In this particular example, the average horizontal shift distance is calculated to be approximately 1.5ft, which corresponds very close to the real world horizontal distance between the two centroid points in Figure 41. This result essentially insinuates that although the trajectories produced by the automated detection systems might not be the ground truth, it is still sufficient enough to use them as an indicator where and when the lane changing has occurred. However if the counting zones were to be placed in the same lane transition zone, the counting of the vehicle may falsely be added to the adjacent lane.

Spillover effect also takes part in the proposed occlusion handling techniques, which assumes that the vehicle move in constant speed during the occlusion without any horizontal movement. This assumption however, is not always true as shown in Figure 40 where the detected vehicle weaves after the occlusion has been detected. Although the overall count data will be conserved, this phenomenon cannot be deal by the current framework and count will be added to the lane before lane changing.

Figure 41: Weaving during an Occlusion

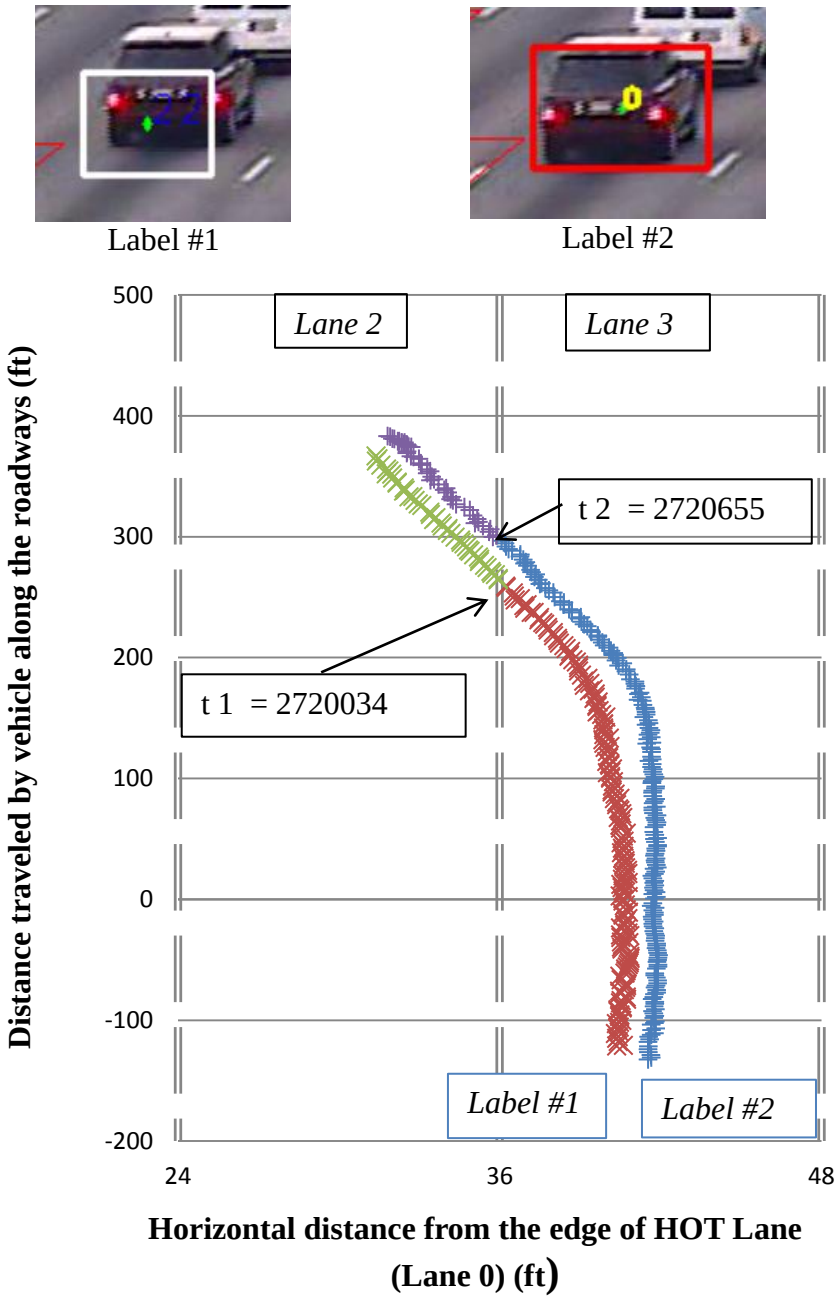


Figure 42: **The Comparison of Trajectory of the Same Vehicle by the Automated Detection (Top Left) and by the Manual Initialization (Top Right). Note: The vehicle IDs given to the same vehicle above are different because both tests were run on separate instances.**

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

In this thesis, the detection-and-tracking framework were developed and tested for feasibility in counting vehicles application, which were made possible by the optimization of the Haar-Cascade implementation. Despite challenges observed during the course of the experiment as explained in the later part of Chapter 7, the proposed framework is demonstrated to be effective as a reliable vehicle counting tool as a part of a vision-based traffic surveillance systems. With few exceptions, the Correct Counting Rate (CCR) with an appropriate minimum neighbor value can attain respectably high counting rate while minimizing the False Count Rate (FCR) in the entire duration of test samples. The systems also seem to exhibit reasonable robustness against changing illumination and object sizes.

The identification and removal of falsely detected object found a considerable success discarding most of the non-vehicle detections out of the counting. Although there were some noticeable errors caused by various traffic characteristics (e.g. congestion level) and driving behavior of an individual vehicle, the occurrences of such errors did not significantly affect the overall Removal Rate (RR) of filtering process, as they were only a small percentage of the total traffic volume. The reliability of proposed framework may further improve by reinforcing the filtering operation with more advanced microscopic traffic theory in the aforementioned challenging environments.

Despite a considerable success in filtering process, there were few issues during the experiments that cannot be neglected for performance evaluation. Contrary to the preliminary assumption, the spillover effects seem to take a huge role in dealing with lane-by-lane statistics of traffic counts. Although the total vehicle counts remain the same, a vehicle count may be added in the adjacent lane based on the accuracy of automated detection. Additionally, since this thesis only concerned the effectiveness of the occlusion handling in terms of false positives which can be detrimental in the proposed counting vehicle framework, the feasibility of this occlusion handling technique may remain questionable. The system accidently missed some occluded vehicles and removed them from counting. This result insinuate that the occlusion handling based on the relative position, scale data and use of acceleration data, although it improves the count as demonstrated throughout this thesis, may not be the best detector for recognizing occlusion. Thus, the author recommends the use of template-based detector to identify an occlusion in addition to the method provided in this thesis for future research. Still, the system demonstrates that the used occlusion handling does contribute to the overall reliability of the system.

Nevertheless, the results provided in this thesis demonstrates that the counting methodology in the traditional vision-based traffic surveillance systems relying on localized virtual detector can be improved in term of accuracy by implementing the proposed detection-tracking framework. More extensive studies may be conducted in the future to demonstrate the effectiveness of the proposed method in much greater testing scale.

APPENDIX A: CODE USED FOR CALIBRATION METHOD

This code is the implementation of the calibration method explained in Kanhere et al. (2010).

```
double k, k_v, del;
double vf, vb;

/* find the vanishing point (u0, v0) */
LineSegment2DF line1 = new LineSegment2DF(imgCoord[0], imgCoord[1]);
LineSegment2DF line2 = new LineSegment2DF(imgCoord[2], imgCoord[3]);
PointF vp = new PointF(); //vanishing point
if (DoLinesIntersect(line1, line2, ref vp) == -1)
{
    GygaxExceptionHandling.GygaxException ge
     = new GygaxExceptionHandling.GygaxException("cannot find
     The vanishing piont!");
    return false;
}
//translation to have a coordinate with an origin at the image center
u0 = vp.X - width / 2;
v0 = vp.Y - height / 2;

/* Y coordinates of the end points of the known length */
vf = imgCoord[4].Y - height / 2;
vb = imgCoord[5].Y - height / 2;

/* the intersection points of middle horizontal line and the line1 (or
line2) corresponding to the known width */

PointF u2 = new PointF();
PointF u3 = new PointF();
LineSegment2DF line3 = new LineSegment2DF(new PointF(0, height / 2),
new PointF(width, height / 2));
DoLinesIntersect(line1, line3, ref u2);
DoLinesIntersect(line2, line3, ref u3);
del = Math.Abs(u2.X - u3.X);

k = (vf - v0) * (vb - v0) / (vf - vb);
k_v = del * k * l / w / v0;

double a = 1.0;
double b = 2 * (u0 * u0 + v0 * v0) - k_v * k_v;
double c = (u0 * u0 + v0 * v0) * (u0 * u0 + v0 * v0) - k_v * k_v * v *
v0;
double discriminant = b * b - 4 * a * c;

if (discriminant < 0)
{
    System.Windows.MessageBox.Show("The discriminant is negative! Try
    to adjust points.");
    return false;
```

```
}

double mp = (-b + Math.Sqrt(discriminant)) / 2 / a;
double mn = (-b - Math.Sqrt(discriminant)) / 2 / a;

if(mp > 0)
    f = Math.Sqrt(mp);
else if(mn > 0)
    f = Math.Sqrt(mn);

phi = Math.Atan(-v0 / f);
double sp = Math.Sin(phi);
double cp = Math.Cos(phi);

theta = Math.Atan(-u0 * Math.Cos(phi) / f);
direction = (line1.Direction.Y > 0) ? 1 : -1;

h = f * w * sp / del / Math.Cos(theta);

T = new Matrix<double>[z.Count];
invT = new Matrix<double>[z.Count];
for (int i = 0; i < z.Count; i++)
{
    invT[i] = new Matrix<double>(3, 3);
    invT[i].Data[0, 0] = f;
    invT[i].Data[1, 1] = -f * sp;
    invT[i].Data[1, 2] = f * (h - z[i]) * cp;
    invT[i].Data[2, 1] = cp;
    invT[i].Data[2, 2] = (h - z[i]) * sp;

    T[i] = new Matrix<double>(3, 3);
    CvInvoke.cvInvert(invT[i].Ptr, T[i].Ptr,
    Emgu.CV.CvEnum.INVERT_METHOD.CV_SVD);
}
rotate_theta.SetRotation(new System.Drawing.PointF(0, 0), theta * 180
/ Math.PI - ((direction > 0) ? 180 : 0), 1.0);

//Calculating the real world coordinates of the origin point
Matrix<double> p0 = new Matrix<double>(3, 1);
p0.Data[0, 0] = imgCoord[0].X - width / 2;
p0.Data[1, 0] = imgCoord[0].Y - height / 2;
p0.Data[2, 0] = 1.0;

r0 = T[0] * p0;

r0.Data[0, 0] /= r0.Data[2, 0];
r0.Data[1, 0] /= r0.Data[2, 0];
r0.Data[2, 0] = 1;

r0 = rotate_theta * r0;

WriteResult(width, height, inputFileName);

return true;
```

APPENDIX B: LANE-BY-LANE RESULTS FOR ALL SAMPLES

SAMPLE 1

Table 5: Raw Counting Data for Sample 1 for Minimum Neighbor = 1

True (Manual) Count		291	608	504	447	449	436	311	3046
Lane		0	1	2	3	4	5	6	Total
Count		246	589	521	478	481	436	300	3051
Over-Count	Double Count	1	5	6	26	23	17	1	79
	Falsely Counted	1	6	15	22	13	3	3	63
	Falsely Added to Counter by OH	0	0	2	3	3	0	0	8
Under Count	Falsely Removed by Filter	8	2	1	1	1	1	4	18
False Occlusion (Correct Count)		0	1	8	7	9	2	0	27
Correct Occlusion count		0	1	4	5	1	1	0	12
Total Correct Vehicle		244	578	498	427	442	416	296	2901
Total False Count		2	11	23	51	39	20	4	150
Total Missed Vehicle		47	30	6	20	7	20	15	145
Total Removed		14	53	284	287	244	87	16	985
Total Correct Add by OH		0	2	12	12	10	3	0	39
CCR (%)		83.8%	95.1%	98.8%	95.5%	98.4%	95.4%	95.2%	95.2%
FCR (%)		0.8%	1.9%	4.4%	10.7%	8.1%	4.6%	1.3%	4.9%
RR (%)		42.9%	96.2%	99.6%	99.7%	99.6%	98.9%	75.0%	98.2%
OHR (%)		NA	50.0%	92.3%	92.3%	90.9%	75.0%	NA	68.4%

Table 6: Raw Counting Data for Sample 1 for Minimum Neighbor = 2

True (Manual) Count		291	608	504	447	449	436	311	3046
Lane		0	1	2	3	4	5	6	Total
Count		189	554	495	454	455	431	297	2875
Over-Count	Double Count	0	0	2	8	4	8	0	22
	Falsely Counted	0	2	12	13	4	4	0	35
	Falsely Added to Counter by OH	0	0	2	2	4	1	0	9
Under Count	Falsely Removed by Filter	23	2	6	4	3	1	1	40
False Occlusion (Correct Count)		0	1	4	6	1	3	0	15
Correct Occlusion count		0	0	2	6	6	0	0	14
Total Correct Vehicle		189	552	479	431	443	418	297	2809
Total False Count		0	2	16	23	12	13	0	66
Total Missed Vehicle		102	56	25	16	6	18	14	237
Total Removed		26	25	227	244	165	51	11	749
Total Correct Add by OH		0	1	6	12	7	3	0	29
CCR (%)		64.9%	90.8%	95.0%	96.4%	98.7%	95.9%	95.5%	92.2%
FCR (%)		0.0%	0.4%	3.2%	5.1%	2.6%	3.0%	0.0%	2.3%
RR (%)		11.5%	92.0%	97.4%	98.4%	98.2%	98.0%	90.9%	94.7%
OHR (%)		NA	100.0%	75.0%	85.7%	63.6%	75.0%	NA	76.3%

Table 7: Raw Counting Data for Sample 1 for Minimum Neighbor = 3

True (Manual) Count		291	608	504	447	449	436	311	3046
Lane		0	1	2	3	4	5	6	Total
Count		152	509	474	428	432	407	287	2689
Over-Count	Double Count	0	0	0	3	1	1	0	5
	Falsely Counted	0	1	7	3	0	1	0	12
	Falsely Added to Counter by OH	0	0	0	3	2	0	0	5
Under Count	Falsely Removed by Filter	1	0	2	5	2	1	1	12
False Occlusion (Correct Count)		0	0	0	3	0	1	0	4
Correct Occlusion count		0	3	2	1	2	0	0	8
Total Correct Vehicle		152	508	467	419	429	405	287	2667
Total False Count		0	1	7	9	3	2	0	22
Total Missed Vehicle		139	99	30	19	17	29	24	357
Total Removed		1	4	75	91	57	13	9	250
Total Correct Add by OH		0	3	2	4	2	1	0	12
CCR (%)		52.2%	83.6%	92.7%	93.7%	95.5%	92.9%	92.3%	87.6%
FCR (%)		0.0%	0.2%	1.5%	2.1%	0.7%	0.5%	0.0%	0.8%
RR (%)		0.0%	100.0%	97.3%	94.5%	96.5%	92.3%	88.9%	95.2%
OHR (%)		NA	100.0%	100.0%	57.1%	50.0%	100.0%	NA	70.6%

Table 8: Raw Counting Data for Sample 1 for Minimum Neighbor = 4

True (Manual) Count		291	608	504	447	449	436	311	3046
Lane		0	1	2	3	4	5	6	Total
Count		114	482	453	421	426	399	262	2557
Over-Count	Double Count	0	0	0	3	0	1	0	4
	Falsely Counted	0	0	0	3	0	0	0	3
	Falsely Added to Counter by OH	0	0	0	3	0	0	0	3
Under Count	Falsely Removed by Filter	0	0	3	1	3	1	4	12
False Occlusion (Correct Count)		2	2	2	1	1	0	0	8
Correct Occlusion count		0	0	2	6	1	0	0	9
Total Correct Vehicle		114	482	453	412	426	398	262	2547
Total False Count		0	0	0	9	0	1	0	10
Total Missed Vehicle		177	126	51	35	23	38	49	499
Total Removed		0	1	45	49	26	8	8	137
Total Correct Add by OH		2	2	4	7	2	0	0	17
CCR (%)		39.2%	79.3%	89.9%	92.2%	94.9%	91.3%	84.2%	83.6%
FCR (%)		0.0%	0.0%	0.0%	2.1%	0.0%	0.3%	0.0%	0.4%
RR (%)		NA	100.0%	93.3%	98.0%	88.5%	87.5%	50.0%	91.2%
OHR (%)		100.0%	100.0%	100.0%	70.0%	100.0%	NA	NA	85.0%

Table 9: Raw Counting Data for Sample 1 for Minimum Neighbor = 5

True (Manual) Count		291	608	504	447	449	436	311	3046
Lane		0	1	2	3	4	5	6	Total
Count		84	448	440	406	411	385	226	2400
Over Count	Double Count	0	0	0	1	1	0	0	2
	Falsely Counted	0	0	1	1	0	1	0	3
	Falsely Added to Counter by OH	0	0	1	1	0	0	0	2
Under Count	Falsely Removed by Filter	2	1	1	2	4	0	4	14
False Occlusion (Correct Count)		0	0	1	3	2	2	0	8
Correct Occlusion count		0	1	0	2	1	0	0	4
Total Correct Vehicle		84	448	438	403	410	384	226	2393
Total False Count		0	0	2	3	1	1	0	7
Total Missed Vehicle		177	126	51	35	23	38	49	499
Total Removed		2	1	27	26	20	5	7	88
Total Correct Add by OH		0	1	1	5	3	2	0	12
CCR (%)		28.9%	73.7%	86.9%	90.2%	91.3%	88.1%	72.7%	78.6%
FCR (%)		0.0%	0.0%	0.5%	0.7%	0.2%	0.3%	0.0%	0.3%
RR (%)		0.0%	0.0%	96.3%	92.3%	80.0%	100.0%	42.9%	84.1%
OHR (%)		NA	100.0%	50.0%	83.3%	100.0%	100.0%	NA	85.7%

SAMPLE 2

Table 10: Raw Counting Data for Sample 2 for Minimum Neighbor = 1

True (Manual) Count		260	353	314	369	254	1550
Lane		0	1	2	3	4	Total
Count		245	352	326	352	251	1526
Over Count	Double Count	1	23	16	11	2	53
	Falsely Counted	0	4	16	4	3	27
	Falsely Added to Counter by OH	0	1	0	0	0	1
Under Count	Falsely Removed by Filter	6	6	12	5	5	34
False Occlusion (Correct Count)		2	1	0	1	3	7
Correct Occlusion count		0	1	2	0	0	3
Total Correct Vehicle		244	324	294	337	246	1445
Total False Count		1	28	32	15	5	81
Total Missed Vehicle		16	29	20	32	8	105
Total Removed		8	42	157	123	11	341
Total Correct Add by OH		2	2	2	1	3	10
CCR (%)		93.8%	91.8%	93.6%	91.3%	96.9%	93.2%
FCR (%)		0.4%	8.0%	9.8%	4.3%	2.0%	5.3%
RR (%)		25.0%	85.7%	92.4%	95.9%	54.5%	90.0%
OHR (%)		100.0%	66.7%	100.0%	100.0%	100.0%	90.9%

Table 11: Raw Counting Data for Sample 2 for Minimum Neighbor = 2

True (Manual) Count		260	353	314	369	254	1550
Lane		0	1	2	3	4	Total
Count		230	339	305	346	235	1455
Over-Count	Double Count	0	16	11	1	0	28
	Falsely Counted	0	4	4	1	0	9
	Falsely Added to Counter by OH	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	3	12	8	6	7	36
False Occlusion (Correct Count)		0	0	1	0	0	1
Correct Occlusion count		0	1	2	0	0	3
Total Correct Vehicle		230	319	290	344	235	1418
Total False Count		0	20	15	2	0	37
Total Missed Vehicle		30	34	24	25	19	132
Total Removed		8	25	44	46	10	133
Total Correct Add by OH		0	1	3	0	0	4
CCR (%)		88.5%	90.4%	92.4%	93.2%	92.5%	91.5%
FCR (%)		0.0%	5.9%	4.9%	0.6%	0.0%	2.5%
RR (%)		62.5%	52.0%	81.8%	87.0%	30.0%	72.9%
OHR (%)		NA	100.0%	100.0%	NA	NA	100.0%

Table 12: Raw Counting Data for Sample 2 for Minimum Neighbor = 3

True (Manual) Count		260	353	314	369	254	1550
Lane		0	1	2	3	4	Total
Count		216	318	281	335	231	1381
Over-Count	Double Count	0	2	4	3	1	10
	Falsely Counted	0	0	0	0	0	0
	Falsely Added to Counter by OH	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	3	9	8	6	8	34
False Occlusion (Correct Count)		0	0	1	1	0	2
Correct Occlusion count		0	1	1	0	0	2
Total Correct Vehicle		216	316	277	332	230	1371
Total False Count		0	2	4	3	1	10
Total Missed Vehicle		44	37	37	37	24	179
Total Removed		4	17	19	27	10	77
Total Correct Add by OH		0	1	2	1	0	4
CCR (%)		83.1%	89.5%	88.2%	90.0%	90.6%	88.5%
FCR (%)		0.0%	0.6%	1.4%	0.9%	0.4%	0.7%
RR (%)		25.0%	47.1%	57.9%	77.8%	20.0%	55.8%
OHR (%)		NA	100.0%	100.0%	100.0%	NA	100.0%

Table 13: Raw Counting Data for Sample 2 for Minimum Neighbor = 4

True (Manual) Count		260	353	314	369	254	1550
Lane		0	1	2	3	4	Total
Count		203	305	266	329	222	1325
Over-Count	Double Count	0	3	2	1	0	6
	Falsely Counted	0	1	0	0	0	1
	Falsely Added to Counter by OH	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	1	9	7	4	8	29
False Occlusion (Correct Count)		0	0	1	0	0	1
Correct Occlusion count		0	1	3	0	0	4
Total Correct Vehicle		203	301	264	328	222	1318
Total False Count		0	4	2	1	0	7
Total Missed Vehicle		57	52	50	41	32	232
Total Removed		3	15	11	12	8	49
Total Correct Add by OH		0	1	4	0	0	5
CCR (%)		78.1%	85.3%	84.1%	88.9%	87.4%	85.0%
FCR (%)		0.0%	1.3%	0.8%	0.3%	0.0%	0.5%
RR (%)		66.7%	40.0%	36.4%	66.7%	0.0%	40.8%
OHR (%)		NA	100.0%	100.0%	NA	NA	100.0%

Table 14: Raw Counting Data for Sample 2 for Minimum Neighbor = 5

True (Manual) Count		260	353	314	369	254	1550
Lane		0	1	2	3	4	Total
Count		188	294	258	310	215	1265
Over-Count	Double Count	0	2	0	1	0	3
	Falsely Counted	0	1	0	0	0	1
	Falsely Added to Counter by OH	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	5	10	8	3	5	31
False Occlusion (Correct Count)		0	0	0	1	0	1
Correct Occlusion count		0	0	0	0	0	0
Total Correct Vehicle		188	291	258	309	215	1261
Total False Count		0	3	0	1	0	4
Total Missed Vehicle		72	62	56	60	39	289
Total Removed		7	17	14	7	8	53
Total Correct Add by OH		0	0	0	1	0	1
CCR (%)		72.3%	82.4%	82.2%	83.7%	84.6%	81.4%
FCR (%)		0.0%	1.0%	0.0%	0.3%	0.0%	0.3%
RR (%)		28.6%	41.2%	42.9%	57.1%	37.5%	41.5%
OHR (%)		NA	NA	NA	100.0%	NA	100.0%

SAMPLE 3

Table 15: Raw Counting Data for Sample 3 for Minimum Neighbor = 1

True (Manual) Count		415	449	420	383	332	394	274	2667
Lane		0	1	2	3	4	5	6	Total
Count		414	434	431	380	332	395	274	2660
Over-Count	Double Count	0	1	15	14	12	3	5	50
	Falsely Counted	0	1	1	0	0	0	0	2
	Falsely Added to Counter by OH	0	0	2	0	1	0	0	3
Under Count	Falsely Removed by Filter	1	0	0	1	1	0	0	3
False Occlusion (Correct Count)		0	0	0	0	1	0	0	1
Correct Occlusion count		0	0	1	2	0	0	0	3
Total Correct Vehicle		414	432	413	366	319	392	269	2605
Total False Count		0	2	18	14	13	3	5	55
Total Missed Vehicle		1	17	7	17	13	2	5	62
Total Removed		2	8	10	13	1	0	0	34
Total Correct Add by OH		0	0	1	2	1	0	0	4
CCR (%)		99.8%	96.2%	98.3%	95.6%	96.1%	99.5%	98.2%	97.7%
FCR (%)		0.0%	0.5%	4.2%	3.7%	3.9%	0.8%	1.8%	2.1%
RR (%)		50.0%	100.0%	100.0%	92.3%	0.0%	NA	NA	91.2%
OHR (%)		NA	NA	33.3%	100.0%	50.0%	NA	NA	57.1%

Table 16: Raw Counting Data for Sample 3 for Minimum Neighbor = 2

True (Manual) Count		415	449	420	383	332	394	274	2667
Lane		0	1	2	3	4	5	6	Total
Count		409	429	419	369	329	390	267	2612
Over-Count	Double Count	0	2	15	7	8	4	3	39
	Falsely Counted	0	1	0	0	0	0	0	1
	Falsely Added to Counter by OH	0	0	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	1	6	0	0	2	0	0	9
False Occlusion (Correct Count)		0	0	0	0	1	0	0	1
Correct Occlusion count		0	0	1	1	0	0	0	2
Total Correct Vehicle		409	426	404	362	321	386	264	2572
Total False Count		0	3	15	7	8	4	3	40
Total Missed Vehicle		6	23	16	21	11	8	10	95
Total Removed		3	12	4	1	2	0	0	22
Total Correct Add by OH		0	0	1	1	1	0	0	3
CCR (%)		98.6%	94.9%	96.2%	94.5%	96.7%	98.0%	96.4%	96.4%
FCR (%)		0.0%	0.7%	3.6%	1.9%	2.4%	1.0%	1.1%	1.5%
RR (%)		66.7%	50.0%	100.0%	100.0%	0.0%	NA	NA	59.1%
OHR (%)		NA	NA	100.0%	100.0%	100.0%	NA	NA	100.0%

Table 17: Raw Counting Data for Sample 3 for Minimum Neighbor = 3

True (Manual) Count		415	449	420	383	332	394	274	2667
Lane		0	1	2	3	4	5	6	Total
Count		408	431	402	358	321	381	261	2562
Over-Count	Double Count	0	0	4	3	3	1	2	13
	Falsely Counted	0	0	0	0	0	0	0	0
	Falsely Added to Counter by OH	0	0	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	1	0	1	3	0	0	0	5
False Occlusion (Correct Count)		0	0	0	0	0	0	0	0
Correct Occlusion count		0	0	1	2	0	0	0	3
Total Correct Vehicle		408	431	398	355	318	380	259	2549
Total False Count		0	0	4	3	3	1	2	13
Total Missed Vehicle		7	18	22	28	14	14	15	118
Total Removed		2	1	6	11	0	0	0	20
Total Correct Add by OH		0	0	1	2	0	0	0	3
CCR (%)		98.3%	96.0%	94.8%	92.7%	95.8%	96.4%	94.5%	95.6%
FCR (%)		0.0%	0.0%	1.0%	0.8%	0.9%	0.3%	0.8%	0.5%
RR (%)		50.0%	100.0%	83.3%	72.7%	NA	NA	NA	75.0%
OHR (%)		NA	NA	100.0%	100.0%	NA	NA	NA	100.0%

Table 18: Raw Counting Data for Sample 3 for Minimum Neighbor = 4

True (Manual) Count		415	449	420	383	332	394	274	2667
Lane		0	1	2	3	4	5	6	Total
Count		404	427	393	353	318	365	255	2515
Over-Count	Double Count	0	0	1	2	0	0	2	5
	Falsely Counted	0	0	0	0	0	0	0	0
	Falsely Added to Counter by OH	0	0	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	1	3	1	0	0	1	0	6
False Occlusion (Correct Count)		0	0	0	0	0	0	0	0
Correct Occlusion count		0	0	1	3	0	0	0	4
Total Correct Vehicle		404	427	392	351	318	365	253	2510
Total False Count		0	0	1	2	0	0	2	5
Total Missed Vehicle		11	22	28	32	14	29	21	157
Total Removed		1	7	4	0	1	1	0	14
Total Correct Add by OH		0	0	1	3	0	0	0	4
CCR (%)		97.3%	95.1%	93.3%	91.6%	95.8%	92.6%	92.3%	94.1%
FCR (%)		0.0%	0.0%	0.3%	0.6%	0.0%	0.0%	0.8%	0.2%
RR (%)		0.0%	57.1%	75.0%	NA	100.0%	0.0%	NA	57.1%
OHR (%)		NA	NA	100.0%	100.0%	NA	NA	NA	100.0%

Table 19: Raw Counting Data for Sample 3 for Minimum Neighbor = 5

True (Manual) Count		415	449	420	383	332	394	274	2667
Lane		0	1		3	4	5	6	Total
Count		404	431	391	345	313	361	253	2498
Over-Count	Double Count	0	0	0	0	2	0	1	3
	Falsely Counted	0	0	0	0	0	0	0	0
	Falsely Added to Counter by OH	0	0	1	0	0	0	0	1
Under Count	Falsely Removed by Filter	1	0	1	0	0	3	0	5
False Occlusion (Correct Count)		0	0	0	0	0	0	0	0
Correct Occlusion count		0	0	1	4	0	0	0	5
Total Correct Vehicle		404	431	390	345	311	361	252	2494
Total False Count		0	0	1	0	2	0	1	4
Total Missed Vehicle		11	18	30	38	21	33	22	173
Total Removed		1	0	3	0	2	9	0	15
Total Correct Add by OH		0	0	1	4	0	0	0	5
CCR (%)		97.3%	96.0%	92.9%	90.1%	93.7%	91.6%	92.0%	93.5%
FCR (%)		0.0%	0.0%	0.3%	0.0%	0.6%	0.0%	0.4%	0.2%
RR (%)		0.0%	NA	66.7%	NA	100.0%	66.7%	NA	66.7%
OHR (%)		NA	NA	50.0%	100.0%	NA	NA	NA	83.3%

SAMPLE 4

Table 20: Raw Counting Data for Sample 4 for Minimum Neighbor = 1

True (Manual) Count		266	519	454	430	391	370	239	2669
Lane		0	1	2	3	4	5	6	Total
Count		260	515	463	444	394	372	243	2691
Over-Count	Double Count	0	1	6	18	13	3	2	43
	Falsely Counted	0	1	5	1	0	0	2	9
	Falsely Added to Counter by OH	0	0	0	1	0	0	0	1
Under Count	Falsely Removed by Filter	1	4	0	1	0	0	1	7
False Occlusion (Correct Count)		0	0	0	0	0	0	0	0
Correct Occlusion count		0	0	0	0	0	0	0	0
Total Correct Vehicle		260	513	452	424	381	369	239	2638
Total False Count		0	2	11	20	13	3	4	53
Total Missed Vehicle		6	6	2	6	10	1	0	31
Total Removed		2	39	9	2	0	0	2	54
Total Correct Add by OH		0	0	0	0	0	0	0	0
CCR (%)		97.7%	98.8%	99.6%	98.6%	97.4%	99.7%	100.0%	98.8%
FCR (%)		0.0%	0.4%	2.4%	4.5%	3.3%	0.8%	1.6%	2.0%
RR (%)		50.0%	89.7%	100.0%	50.0%	NA	NA	50.0%	87.0%
OHR (%)		NA	NA	NA	0.0%	NA	NA	NA	NA

Table 21: Raw Counting Data for Sample 4 for Minimum Neighbor = 2

True (Manual) Count		266	519	454	430	391	370	239	2669
Lane		0	1	2	3	4	5	6	Total
Count		264	508	447	429	393	371	237	2649
Over-Count	Double Count	0	1	2	13	7	2	0	25
	Falsely Counted	0	0	0	0	0	0	0	0
	Falsely Added to Counter by OH	0	0	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	0	0	0	0	0	0	2	2
False Occlusion (Correct Count)		0	0	0	0	0	0	0	0
Correct Occlusion count		0	0	0	0	0	0	0	0
Total Correct Vehicle		264	507	445	416	386	369	237	2624
Total False Count		0	1	2	13	7	2	0	25
Total Missed Vehicle		2	12	9	14	5	1	2	45
Total Removed		3	11	6	0	1	0	2	23
Total Correct Add by OH		0	0	0	0	0	0	0	0
CCR (%)		99.2%	97.7%	98.0%	96.7%	98.7%	99.7%	99.2%	98.3%
FCR (%)		0.0%	0.2%	0.4%	3.0%	1.8%	0.5%	0.0%	0.9%
RR (%)		100.0%	100.0%	100.0%	NA	100.0%	NA	0.0%	91.3%
OHR (%)		NA	NA	NA	NA	NA	NA	NA	NA

Table 22: Raw Counting Data for Sample 4 for Minimum Neighbor = 3

True (Manual) Count		266	519	454	430	391	370	239	2669
Lane		0	1	2	3	4	5	6	Total
Count		265	508	442	419	387	371	237	2629
Over-Count	Double Count	0	0	1	8	7	2	0	18
	Falsely Counted	0	0	0	0	0	0	0	0
	Falsely Added to Counter by OH	0	0	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	0	0	1	0	1	0	1	3
False Occlusion (Correct Count)		0	0	0	0	0	0	0	0
Correct Occlusion count		0	0	0	0	0	0	0	0
Total Correct Vehicle		265	508	441	411	380	369	237	2611
Total False Count		0	0	1	8	7	2	0	18
Total Missed Vehicle		1	11	13	19	11	1	2	58
Total Removed		2	5	2	0	1	0	1	11
Total Correct Add by OH		0	0	0	0	0	0	0	0
CCR (%)		99.6%	97.9%	97.1%	95.6%	97.2%	99.7%	99.2%	97.8%
FCR (%)		0.0%	0.0%	0.2%	1.9%	1.8%	0.5%	0.0%	0.7%
RR (%)		100.0%	100.0%	50.0%	NA	0.0%	NA	0.0%	72.7%
NA		NA	NA	NA	NA	NA	NA	NA`	NA

Table 23: Raw Counting Data for Sample 4 for Minimum Neighbor = 4

True (Manual) Count		266	519	454	430	391	370	239	2669
Lane		0	1	2	3	4	5	6	Total
Count		265	500	438	415	383	368	237	2606
Over-Count	Double Count	1	0	0	5	4	2	0	12
	Falsely Counted	0	0	0	0	0	0	0	0
	Falsely Added to Counter by OH	0	0	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	0	0	1	0	0	0	1	2
False Occlusion (Correct Count)		0	0	0	0	0	0	0	0
Correct Occlusion count		0	0	0	0	0	0	0	0
Total Correct Vehicle		264	500	438	410	379	366	237	2594
Total False Count		1	0	0	5	4	2	0	12
Total Missed Vehicle		2	19	16	20	12	4	2	75
Total Removed		0	1	3	0	0	0	1	5
Total Correct Add by OH		0	0	0	0	0	0	0	0
CCR (%)		99.2%	96.3%	96.5%	95.3%	96.9%	98.9%	99.2%	97.2%
FCR (%)		0.4%	0.0%	0.0%	1.2%	1.0%	0.5%	0.0%	0.5%
RR (%)		NA	100.0%	66.7%	NA	NA	NA	0.0%	60.0%
OHR (%)		NA	NA	NA	NA	NA	NA	NA	NA

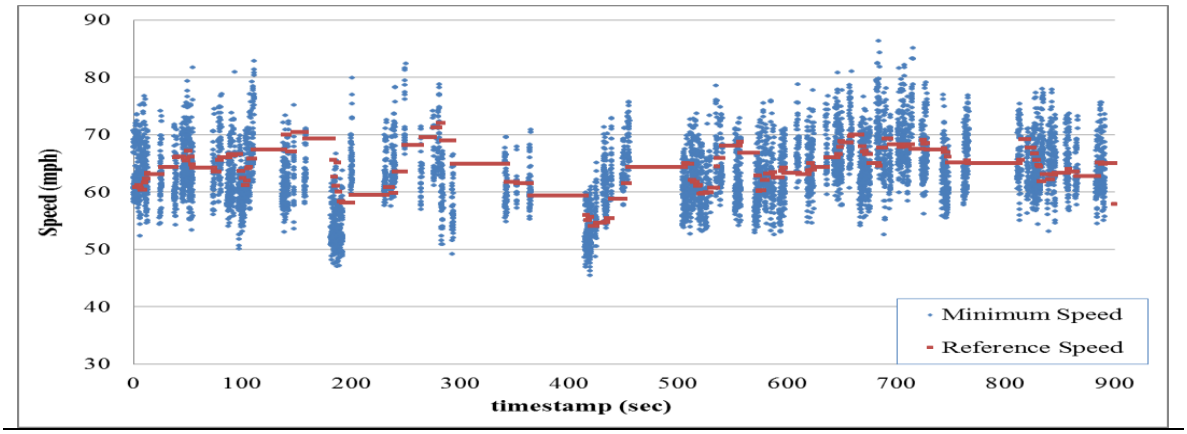
Table 24: Raw Counting Data for Sample 4 for Minimum Neighbor = 5

True (Manual) Count		266	519	454	430	391	370	239	2669
Lane		0	1	2	3	4	5	6	Total
Count		262	469	438	408	377	370	235	2559
Over-Count	Double Count	0	0	0	2	3	0	0	5
	Falsely Counted	0	0	0	0	0	0	0	0
	Falsely Added to Counter by OH	0	0	0	0	0	0	0	0
Under Count	Falsely Removed by Filter	0	0	3	1	0	0	1	5
False Occlusion (Correct Count)		0	0	0	0	0	0	0	0
Correct Occlusion count		0	0	1	0	0	0	0	1
Total Correct Vehicle		262	469	438	406	374	370	235	2554
Total False Count		0	0	0	2	3	0	0	5
Total Missed Vehicle		4	50	16	24	17	0	4	115
Total Removed		0	0	6	3	0	0	2	11
Total Correct Add by OH		0	0	1	0	0	0	0	1
CCR (%)		98.5%	90.4%	96.5%	94.4%	95.7%	100.0%	98.3%	95.7%
FCR (%)		0.0%	0.0%	0.0%	0.5%	0.8%	0.0%	0.0%	0.2%
RR (%)		NA	NA	50.0%	66.7%	NA	NA	50.0%	54.5%
OHR (%)		NA	NA	100.0%	NA	NA	NA	NA	100.0%

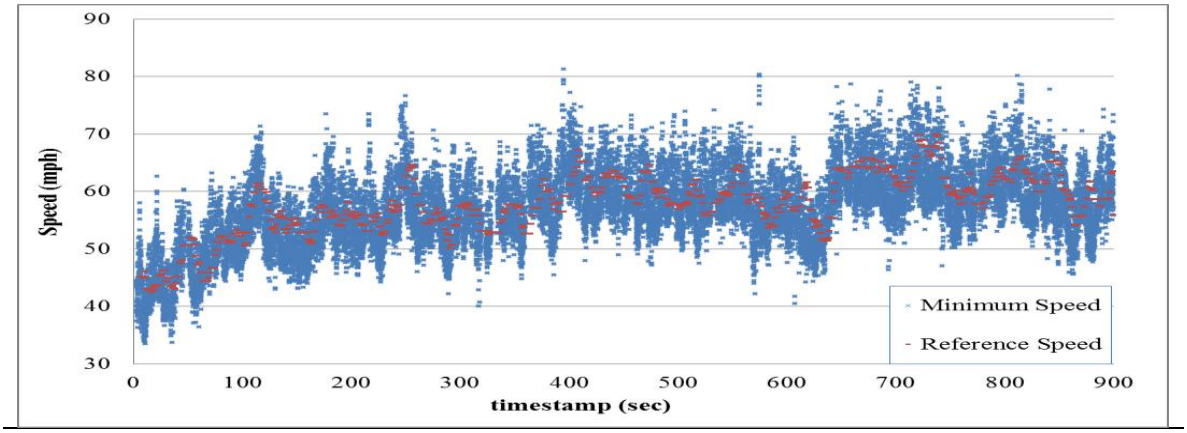
APPENDIX C: REFERENCE SPEED VS. MINIMUM SPEED FOR ALL SAMPLES

SAMPLE 1 (For Min = 3)

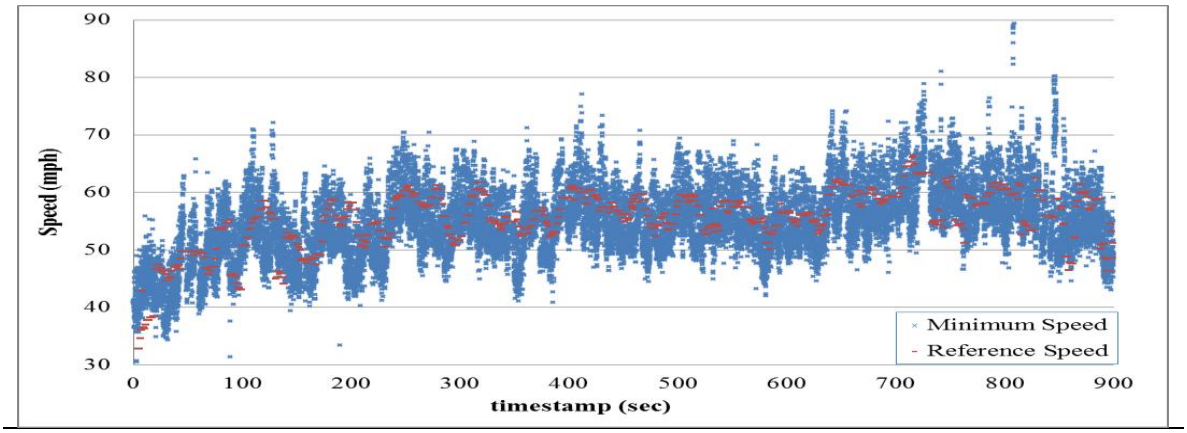
Lane 0



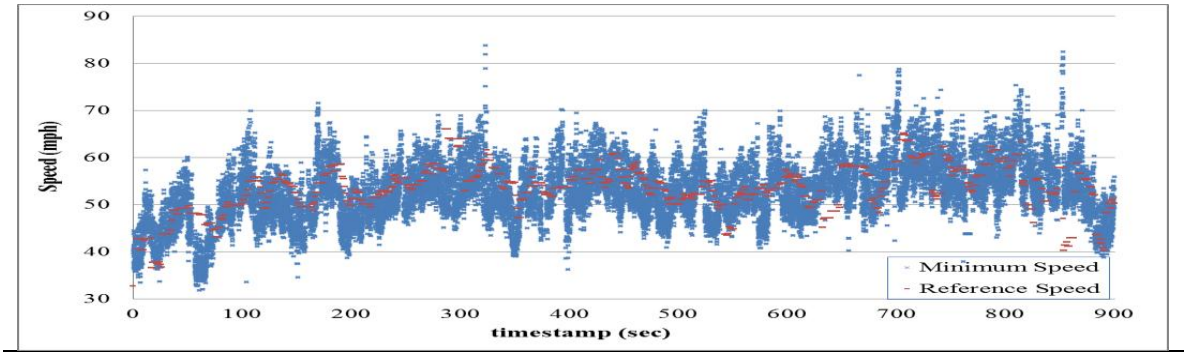
Lane 1



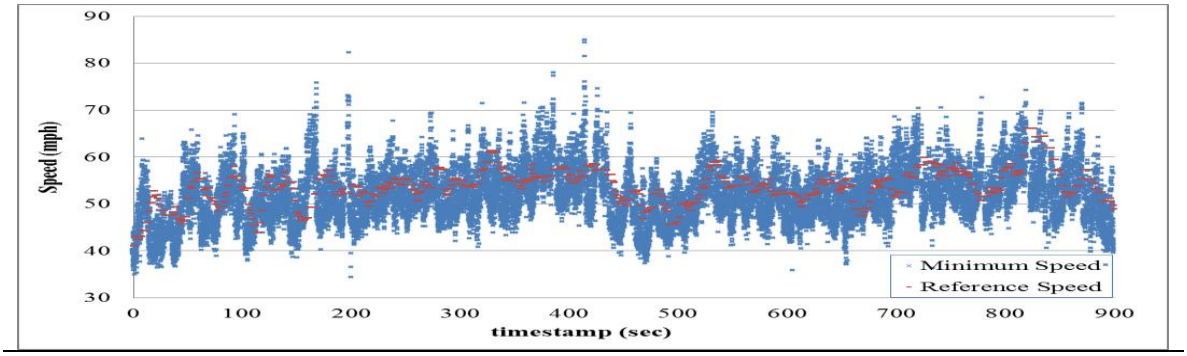
Lane 2



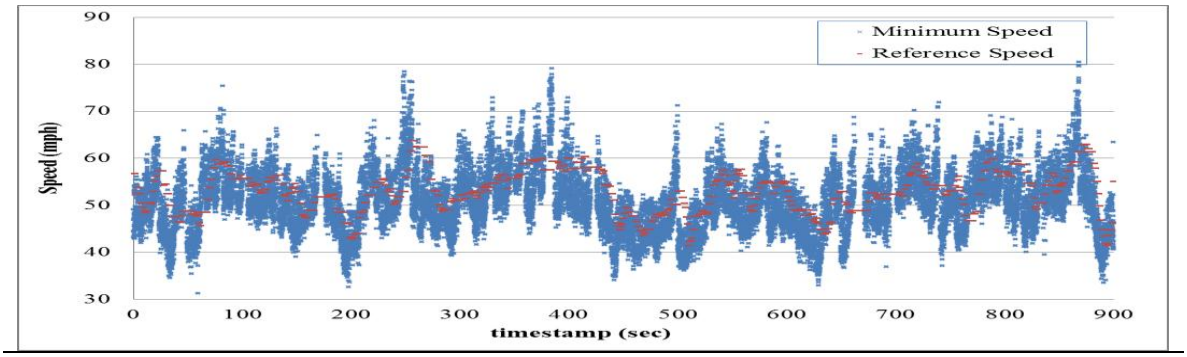
Lane 3



Lane 4



Lane 5



Lane 6

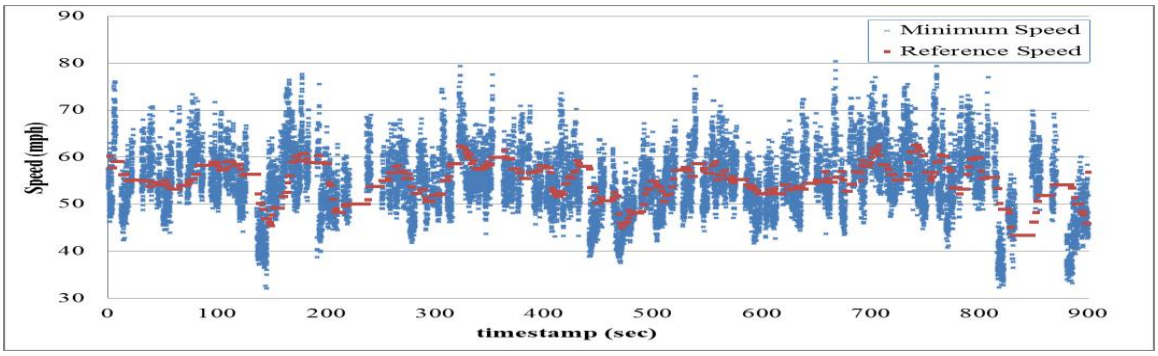


Figure 43: Reference Speed vs. Minimum Speed for All Lanes for Sample 1

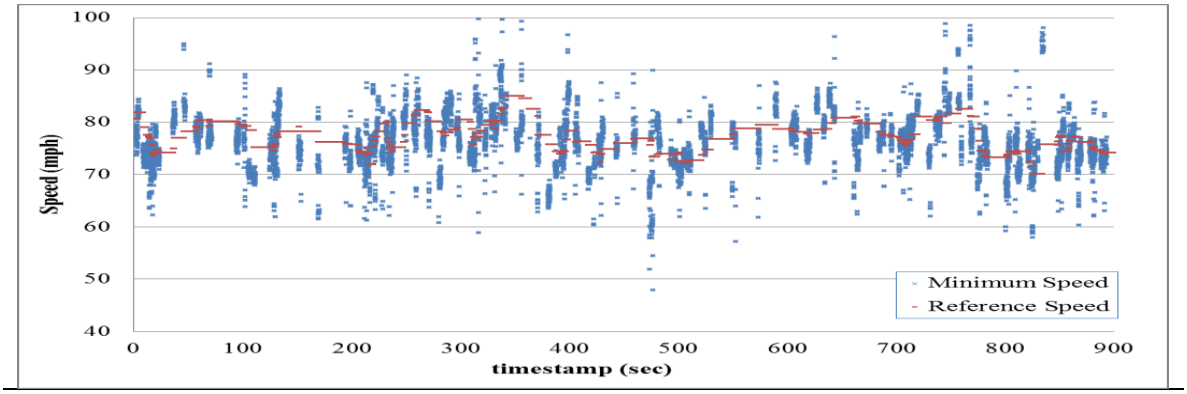
Table 25: Statistics Between Reference Speed and Minimum Speed for All Lanes for Sample 1

Lane	$\mu_{\min-ref}$(mph)	$\sigma_{min-ref}$(mph)	$r_{(min,ref)}$
0	0.650	6.678	0.261
1	0.958	5.576	0.621
2	0.877	5.569	0.557
3	1.115	6.422	0.439
4	2.162	5.448	0.362
5	2.303	5.928	0.460
6	0.181	6.744	0.336

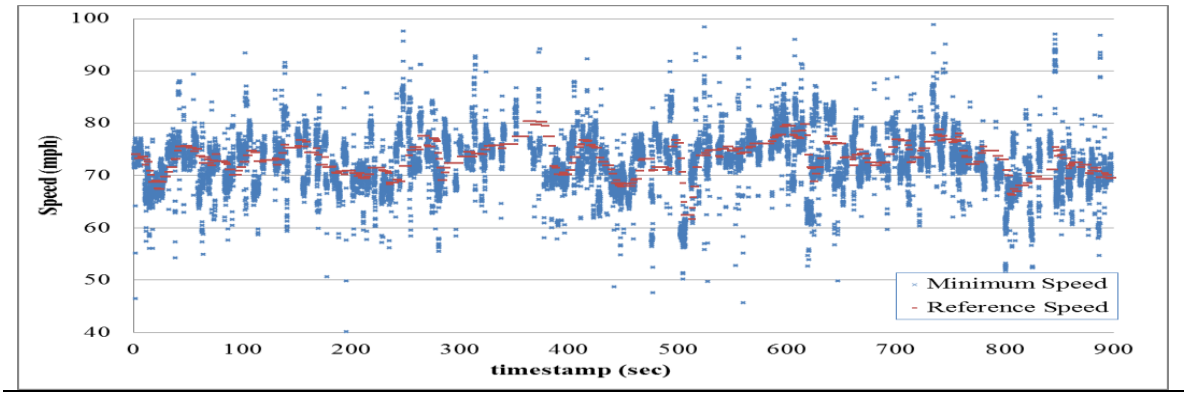
*Note: The calculation is based on the raw data excluding all zero velocities

SAMPLE 2 (For MN = 3)

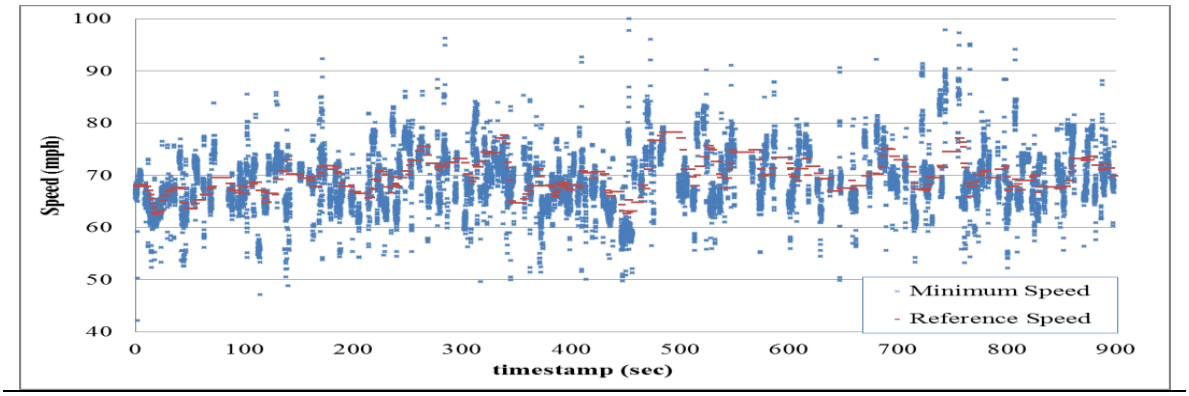
Lane 0



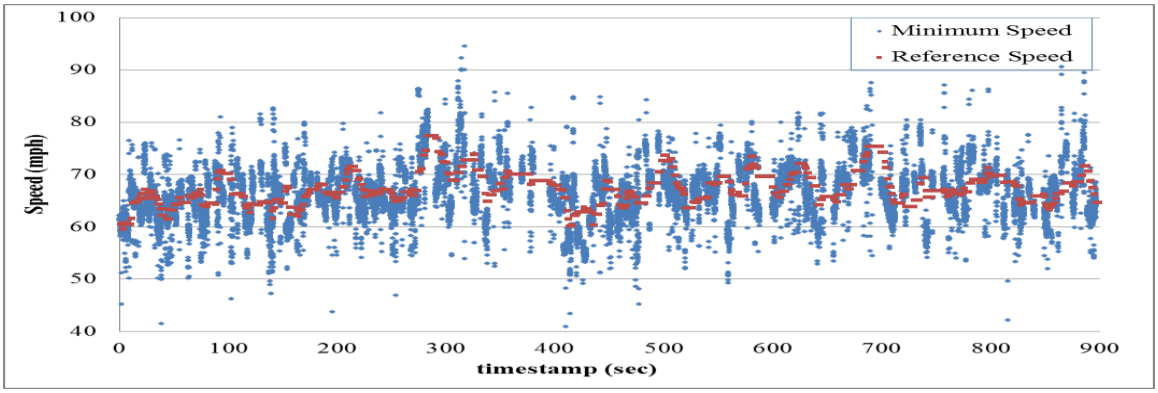
Lane 1



Lane 2



Lane3



Lane4

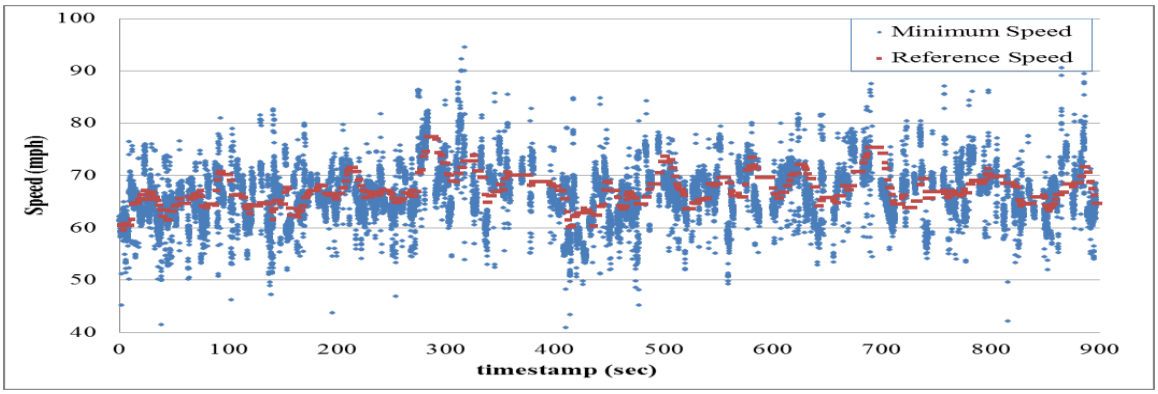


Figure 44: Reference Speed vs. Minimum Speed for All Lanes for Sample 2

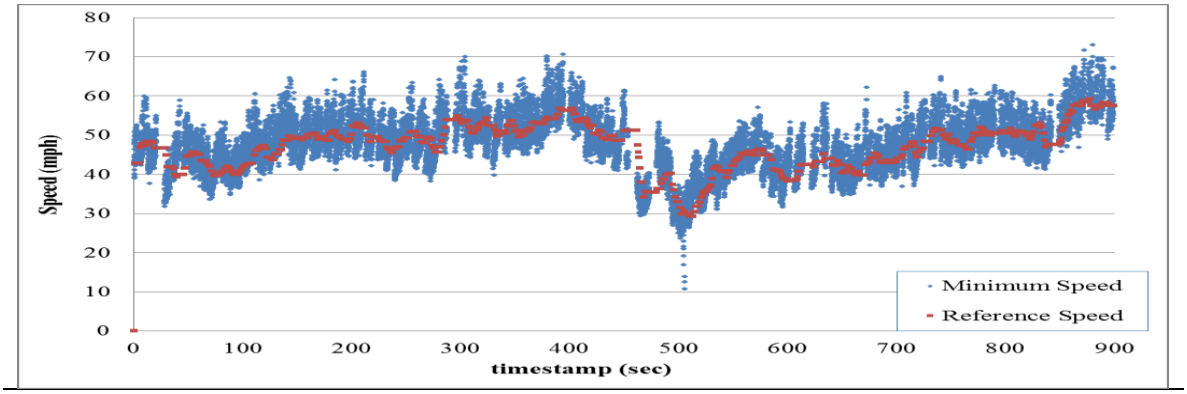
Table 26: Statistics Between Reference Speed and Minimum Speed for All Lanes for Sample 2

Lane	$\mu_{\min-ref}$(mph)	$\sigma_{min-ref}$(mph)	$r_{(min,ref)}$
0	0.973	4.988	0.410
1	0.650	5.478	0.352
2	0.787	5.895	0.314
3	0.808	5.190	0.383
4	1.006	5.606	0.452

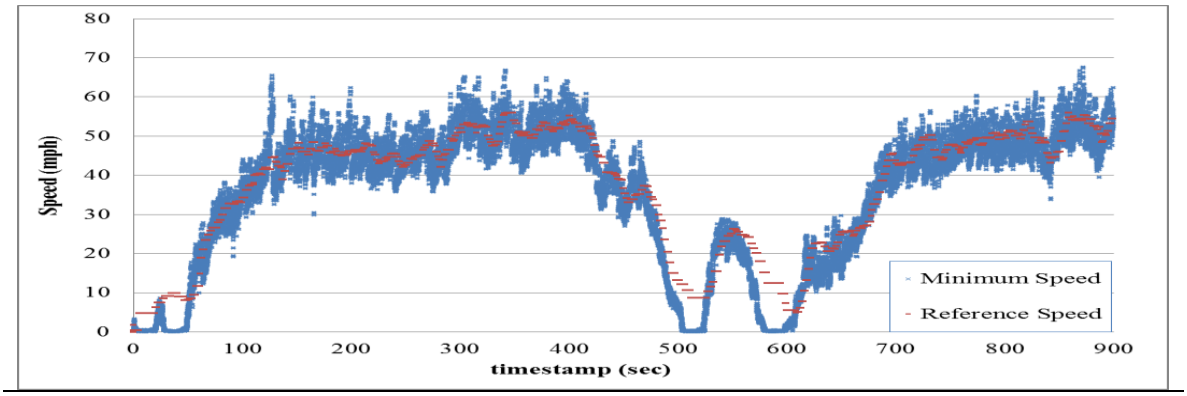
*Note: The calculation is based on the raw data excluding all zero velocities

SAMPLE 3 (For MN = 3)

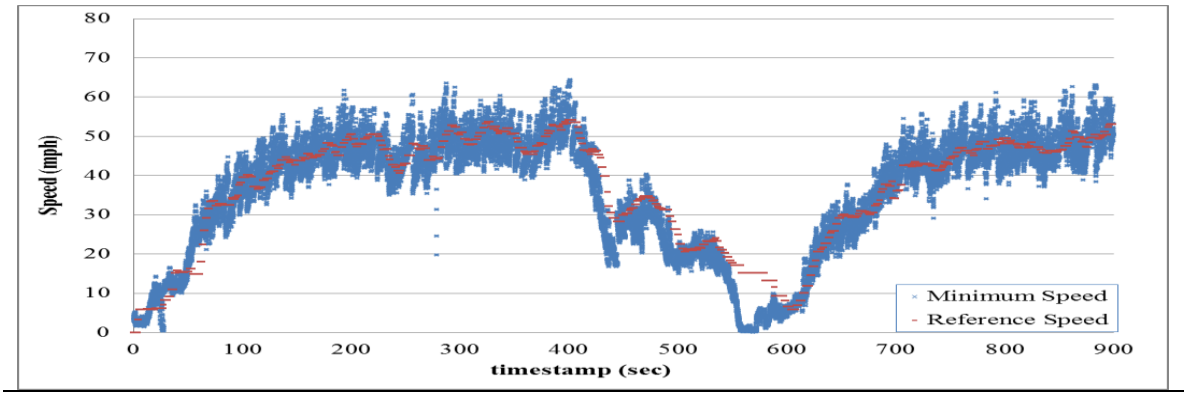
Lane 0



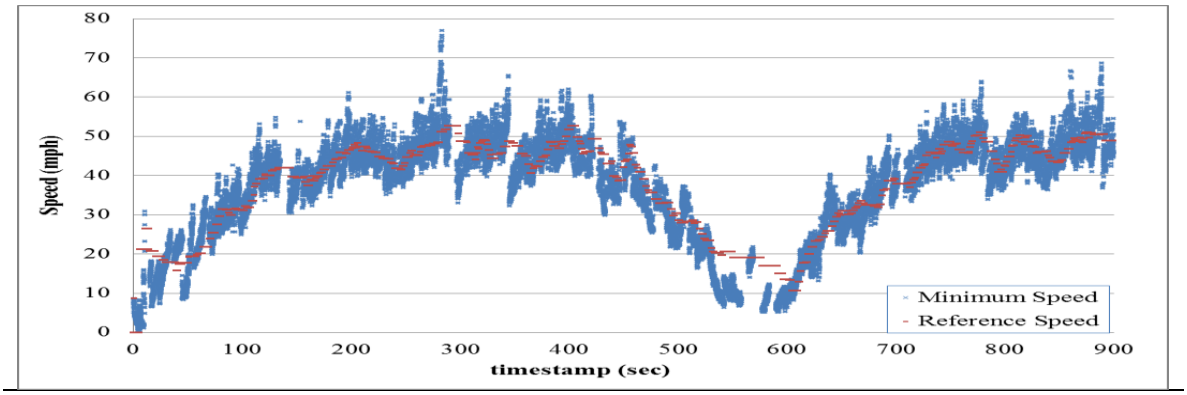
Lane 1



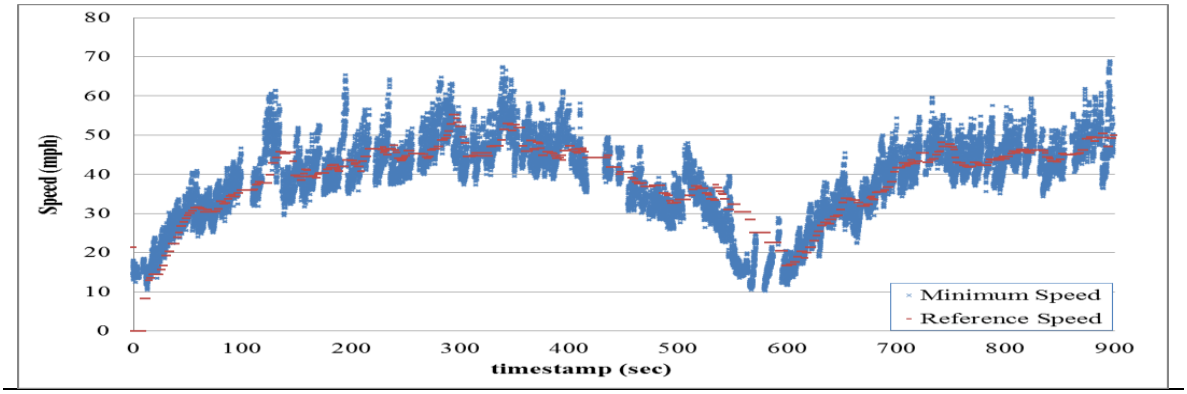
Lane 2



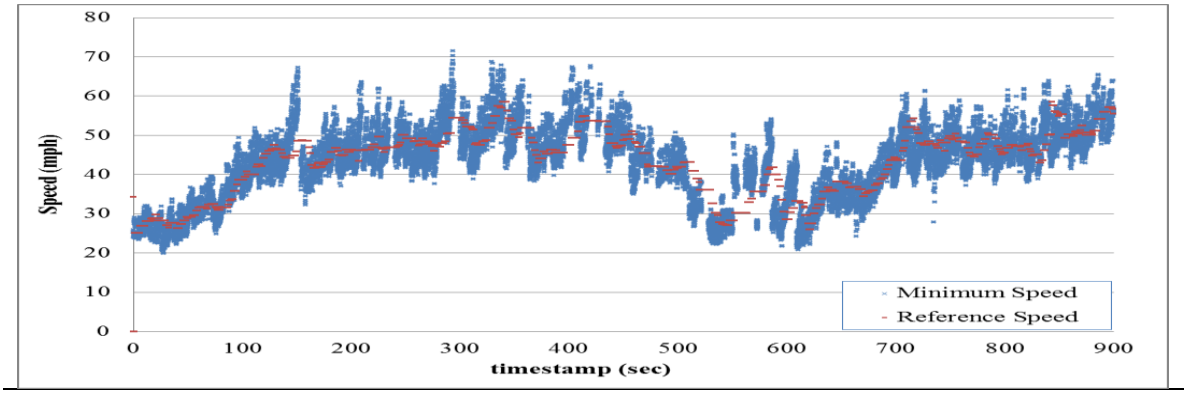
Lane 3



Lane 4



Lane 5



Lane 6

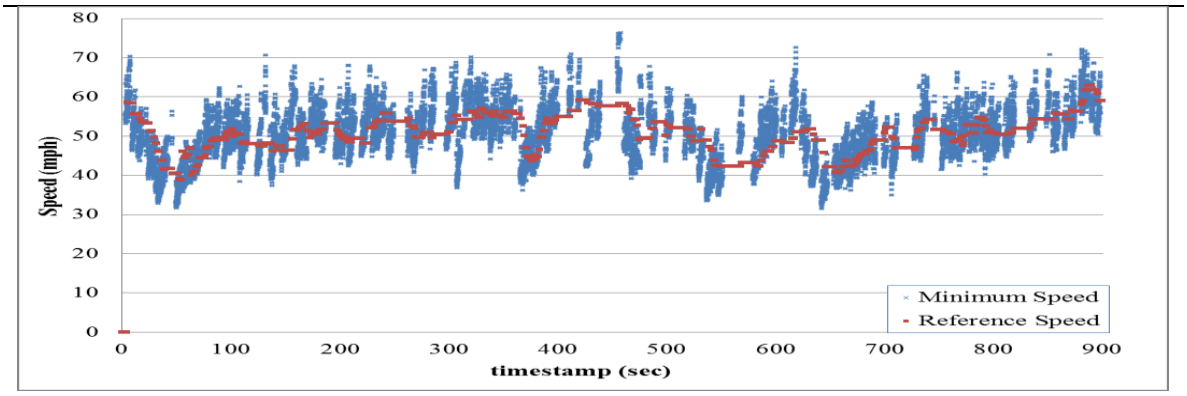


Figure 45: Reference Speed vs. Minimum Speed for All Lanes for Sample 3

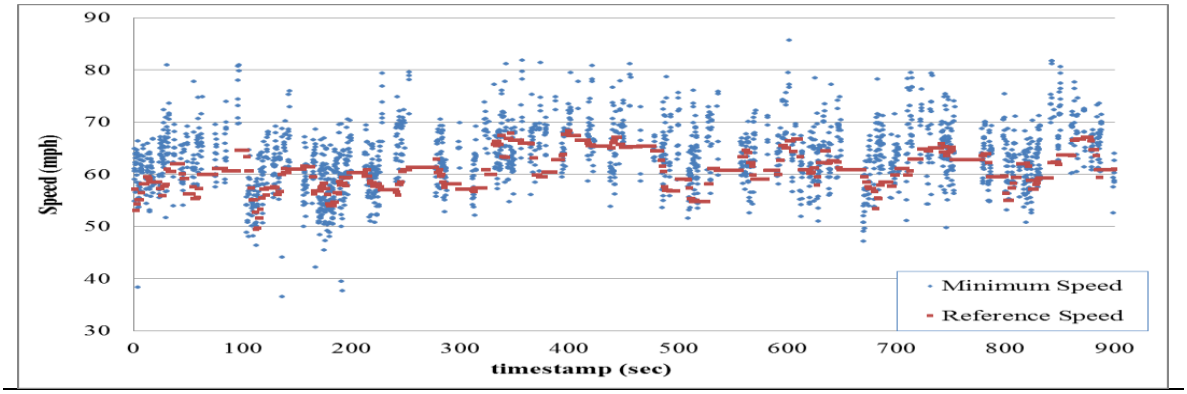
Table 27: Statistics Between Reference Speed and Minimum Speed for All Lanes for Sample 3

Lane	$\mu_{\min-ref}$(mph)	$\sigma_{min-ref}$(mph)	$r_{(min,ref)}$
0	-0.674	4.575	0.762
1	2.001	5.022	0.961
2	1.432	4.988	0.945
3	1.259	5.132	0.921
4	0.139	5.331	0.864
5	0.412	5.174	0.829
6	0.040	6.235	0.484

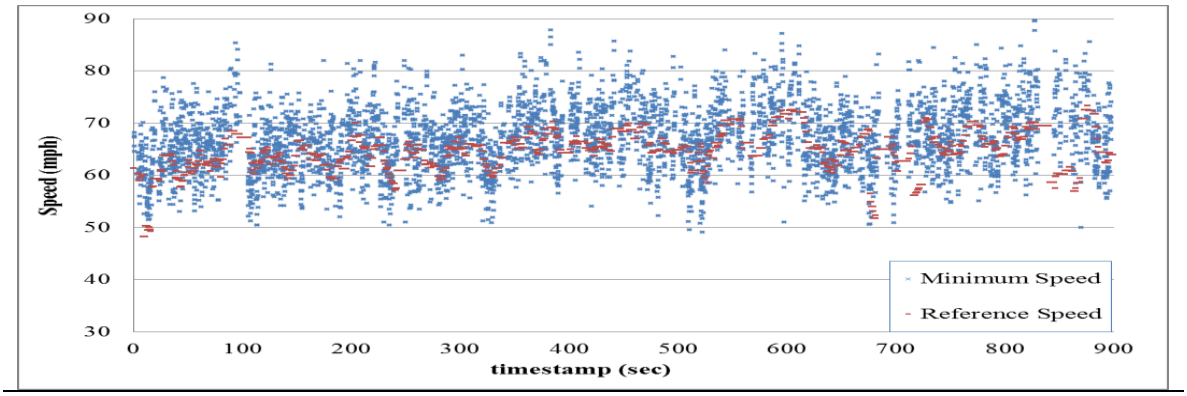
*Note: The calculation is based on the raw data excluding all zero velocities

SAMPLE 4 (For MN = 3)

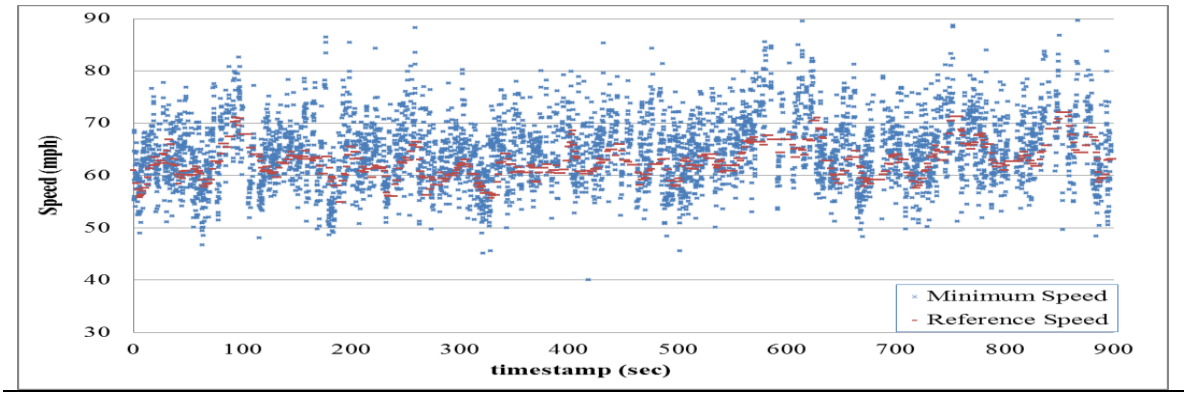
Lane 0



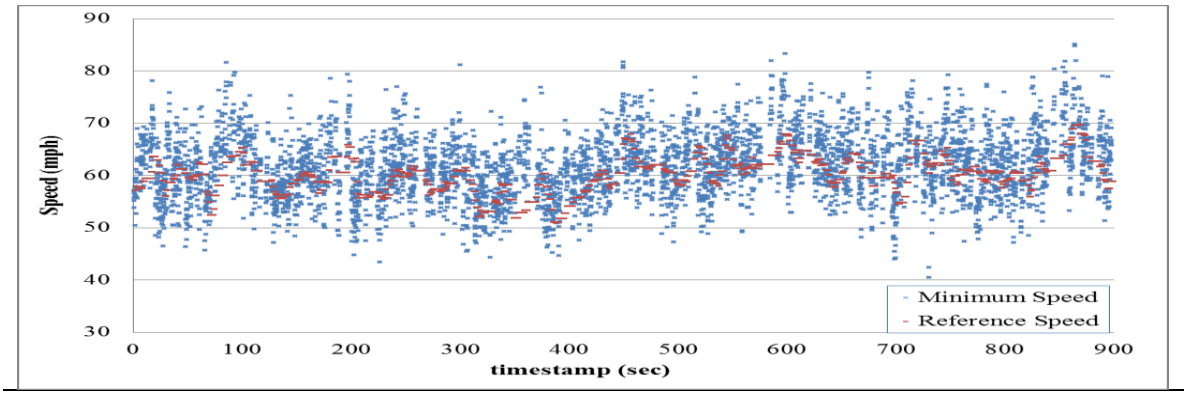
Lane 1



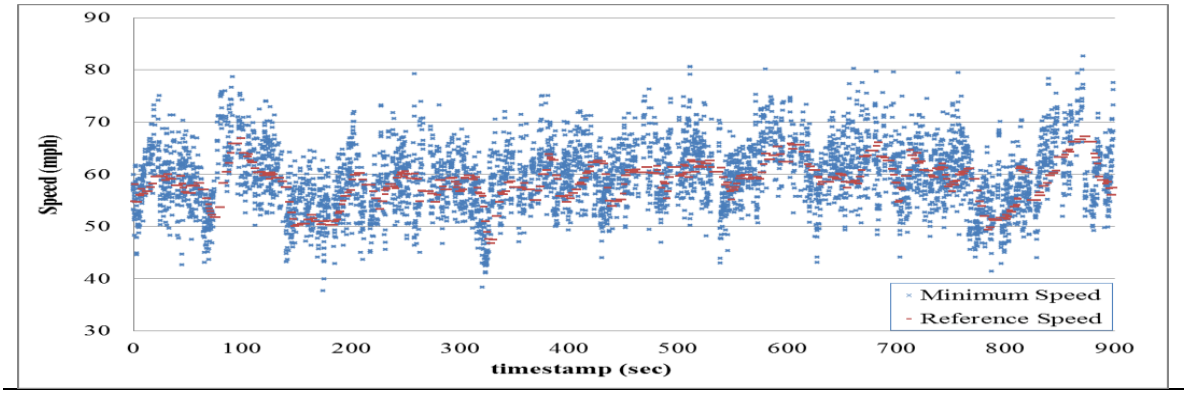
Lane 2



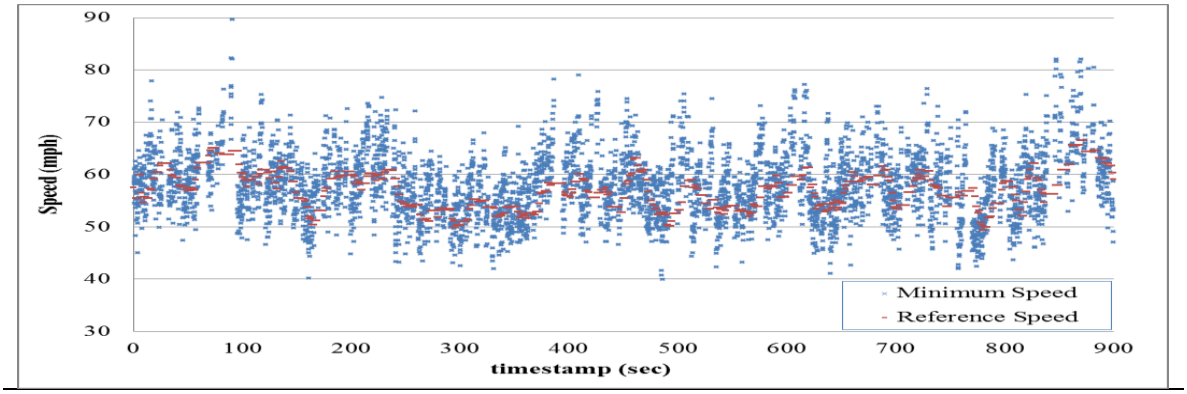
Lane 3



Lane 4



Lane 5



Lane 6

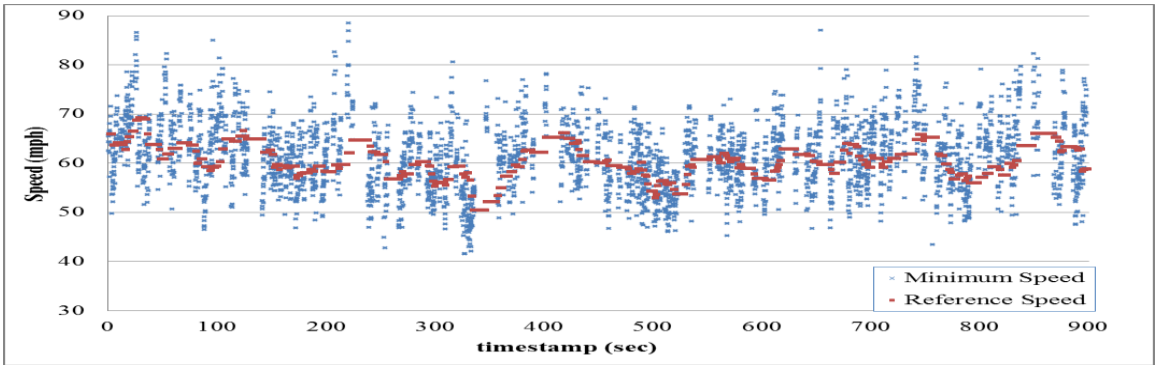


Figure 46: Reference Speed vs. Minimum Speed for All Lanes for Sample 4

Table 28: Statistics Between Reference Speed and Minimum Speed for All Lanes for Sample 4

Lane	$\mu_{\min -ref}$(mph)	$\sigma_{min-ref}$(mph)	$r_{(min,ref)}$
0	-3.008	5.870	0.419
1	-2.358	6.588	0.291
2	-2.292	6.700	0.283
3	-1.528	6.321	0.326
4	-1.043	5.964	0.417
5	-1.136	6.467	0.325
6	-1.041	5.896	0.291

*Note: The calculation is based on the raw data excluding all zero velocities

REFERENCES

"Georgia NaviGAtor." <http://www.georgia-navigator.com/>. (October 15, 2012).

"OpenCV." <http://opencv.org/>. (October 15, 2012).

Avery, R. P., Zhang, G., Wang, Y., and Nihan, N. L. (2007). "Investigation into Shadow Removal from Traffic Images." *Transportation Research Record: Journal of the Transportation Research Board*, 2000(-1), 70-77.

Bas, E., Tekalp, M., and Salman, F. S. (2007). "Automatic vehicle counting from video for traffic flow analysis." Ieee, 392-397.

Boonyanunta, N., and Zeephongsekul, P. (2004). "Predicting the relationship between the size of training sample and the predictive power of classifiers." *Proc., Knowledge-Based Intelligent Information and Engineering Systems*, Springer, 529-535.

Cabrera, R. R., Tuytelaars, T., and Van Gool, L. (2011). "Efficient multi-camera detection, tracking, and identification using a shared set of haar-features." IEEE, 65-71.

Chaiyawatana, N., Uyyanonvara, B., Kondo, T., Dubey, P., and Hatori, Y. "Robust object detection on video surveillance." IEEE, 149-153.

Chang, W. C., and Cho, C. W. (2007). "Multi-class boosting with color-based haar-like features." IEEE, 719-726.

Cheung, S., and Kamath, C. (2007). "Robust techniques for background subtraction in urban traffic video." 881-892.

Crnojevic, V., Antic, B., and Culibrk, D. (2009). "Optimal wavelet differencing method for robust motion detection." IEEE, 645-648.

David, R., Shahram, M., and Waldrop, G. (2012). "Georgia's Regional Signal Operations Program."

Elgammal, A., Duraiswami, R., Harwood, D., and Davis, L. S. (2002). "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance." *Proceedings of the IEEE*, 90(7), 1151-1163.

Enzweiler, M., and Gavrila, D. M. (2009). "Monocular pedestrian detection: Survey and experiments." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12), 2179-2195.

Fang, W., Zhao, Y., Yuan, Y., and Liu, K. (2011). "Real-Time Multiple Vehicles Tracking with Occlusion Handling." *Proc., Image and Graphics (ICIG), 2011 Sixth International Conference on*, IEEE, 667-672.

Feris, R., Petterson, J., Siddiquie, B., Brown, L., and Pankanti, S. (2011). "Large-scale vehicle detection in challenging urban surveillance environments." IEEE, 527-533.

Grant, C., Gillis, B., and Guensler, R. (2000). "Collection of vehicle activity data by video detection for use in transportation planning." *Journal of Intelligent Transportation Systems*, 5(4), 343-361.

Guin, A., Hunter, M., Guensler, R. (2008). "Analysis of Reduction in Effective Capacities of High-Occupancy Vehicle Lanes Related to Traffic Behavior." *Transportation Research Record: Journal of the Transportation Research Board*, 2065(-1), 47-53.

Haselhoff, A., Schauland, S., and Kummert, A. (2008). "A signal theoretic approach to measure the influence of image resolution for appearance-based vehicle detection." *Proc., Intelligent Vehicles Symposium, 2008 IEEE*, IEEE, 822-827.

Hsieh, J. W., Yu, S. H., Chen, Y. S., and Hu, W. F. (2006). "Automatic traffic surveillance system for vehicle tracking and classification." *Intelligent Transportation Systems, IEEE Transactions on*, 7(2), 175-187.

Huang, L. (2010). "Real-time multi-vehicle detection and sub-feature based tracking for traffic surveillance systems." Ieee, 324-328.

Kanhere, N. K. (2008). *Vision-based detection, tracking and classification of vehicles using stable features with automatic camera calibration*, ProQuest.

Kanhere, N. K., and Birchfield, S. T. (2008). "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features." *Intelligent Transportation Systems, IEEE Transactions on*, 9(1), 148-160.

Kanhere, N. K., and Birchfield, S. T. (2010). "A taxonomy and analysis of camera calibration methods for traffic monitoring applications." *Intelligent Transportation Systems, IEEE Transactions on*, 11(2), 441-452.

Kanhere, N. K., Birchfield, S. T., and Sarasua, W. A. (2006). "Vehicle segmentation and tracking in the presence of occlusions." *Transportation Research Record: Journal of the Transportation Research Board*, 1944(-1), 89-97.

Last, M. (2007) "Predicting and Optimizing Classifier Utility with the Power Law." *Proc., Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, IEEE, 219-224.

Lienhart, R., and Maydt, J. (2002) "An extended set of haar-like features for rapid object detection." *Proc., Image Processing. 2002. Proceedings. 2002 International Conference on*, IEEE, I-900-I-903 vol. 901.

Liu, T., Zheng, N., Zhao, L., and Cheng, H. (2005). "Learning based symmetric features selection for vehicle detection." IEEE, 124-129.

Malinovskiy, Y., Wu, Y. J., and Wang, Y. (2009). "Video-Based Vehicle Detection and Tracking Using Spatiotemporal Maps." *Transportation Research Record: Journal of the Transportation Research Board*, 2121(-1), 81-89.

Mehmood, R., Ali, M. U., and Taj, I. A. (2009). "Applying centroid based adjustment to kernel based object tracking for improving localization." IEEE, 209-214.

Messom, C., and Barczak, A. (2006). "Fast and efficient rotated haar-like features using rotated integral images." *Proc., Australasian Conference on Robotics and Automation*, 2.

Mimbela, L. E. Y., and Klein, L. A. (2000). "Summary of vehicle detection and surveillance technologies used in intelligent transportation systems."

Negri, P., Clady, X., and Prevost, L. (2007). "Benchmarking haar and histograms of oriented gradients features applied to vehicle detection." 359-364.

Robert, K. (2009). "Video-based traffic monitoring at day and night vehicle features detection tracking." IEEE, 1-6.

Rodríguez, T., and García, N. (2010). "An adaptive, real-time, traffic monitoring system." *Machine Vision and Applications*, 21(4), 555-576.

Roess, R. P., Prassas, E. S., and McShane, W. R. (2004). *Traffic Engineering*, Prentice-Hall, Inc., Upper Saddle River, New Jersey.

Ross, D. A., Lim, J., Lin, R. S., and Yang, M. H. (2008). "Incremental learning for robust visual tracking." *International Journal of Computer Vision*, 77(1), 125-141.

Sankari, M., and Meena, C. (2011). "Estimation of Dynamic Background and Object Detection in Noisy Visual Surveillance." *International Journal*, 2.

Scharcanski, J., de Oliveira, A. B., Cavalcanti, P. G., and Yari, Y. (2011). "A Particle-Filtering Approach for Vehicular Tracking Adaptive to Occlusions." *Vehicular Technology, IEEE Transactions on*, 60(2), 381-389.

Sheikh, Y., and Shah, M. (2005). "Bayesian modeling of dynamic scenes for object detection." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(11), 1778-1792.

Stauffer, C., and Grimson, W. E. L. (1999). "Adaptive background mixture models for real-time tracking." IEEE.

Tamersoy, B., and Aggarwal, J. (2009). "Robust vehicle detection for tracking in highway surveillance videos using unsupervised learning." Ieee, 529-534.

Thi, H. T. (2007). "A Robust Traffic Surveillance System for Detecting and Tracking Vehicles at Nighttime."University of Technology, Sydney.

Tian, Z., and Kyte, M. (2004) "A special application of a video-image system for vehicle tracking and speed measurement."

Tsai, L. W., Chean, Y. C., Ho, C. P., Gu, H. Z., and Lee, S. Y. (2011). "Multi-Lane Detection and Road Traffic Congestion Classification for Intelligent Transportation System." *Energy Procedia*, 13, 3174-3182.

Tsuchiya, M., and Fujiyoshi, H. (2006). "Evaluating feature importance for object classification in visual surveillance." IEEE, 978-981.

Vargas, M., Milla, J. M., Toral, S. L., and Barrero, F. (2010). "An enhanced background estimation algorithm for vehicle detection in urban traffic scenes." *Vehicular Technology, IEEE Transactions on*, 59(8), 3694-3709.

Viola, P., and Jones, M. (2001)."Rapid object detection using a boosted cascade of simple features." IEEE, I-511-I-518 vol. 511.

Wen, X., Yuan, H., Liu, W., and Zhao, H. (2007). "An improved wavelet feature extraction approach for vehicle detection." IEEE, 1-4.

Yilmaz, A., Javed, O., and Shah, M. (2006). "Object tracking: A survey." *Acm Computing Surveys (CSUR)*, 38(4), 13.

Zhang, G., Avery, R. P., and Wang, Y. (2007). "Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras." *Transportation Research Record: Journal of the Transportation Research Board*, 1993(-1), 138-147.

Zhang, L., Li, S. Z., Yuan, X., and Xiang, S. "Real-time object classification in video surveillance based on appearance learning." IEEE, 1-8.

Zhang, W., Wu, Q. M. J., Yang, X., and Fang, X. (2008). "Multilevel framework to detect and handle vehicle occlusion." *Intelligent Transportation Systems, IEEE Transactions on*, 9(1), 161-174.

Zhang, Z., Li, M., Huang, K., and Tan, T. "Boosting local feature descriptors for automatic objects classification in traffic scene surveillance." IEEE, 1-4.